

Implementazioni di cartografia matematica

Capitolo 1: Il Problema Fondamentale della Rappresentazione Cartografica

1.1 Formulazione Matematica del Problema

La cartografia matematica si confronta con una impossibilità geometrica fondamentale che emerge direttamente dalla teoria delle superfici di Gauss e dalla geometria differenziale riemanniana. La superficie terrestre può essere modellata, in prima approssimazione, come una 2-varietà differenziabile compatta orientabile immersa nello spazio euclideo tridimensionale $\mathbb{R}^3$. Più precisamente, essa è diffeomorfa a una sfera leggermente deformata - un ellissoide di rotazione - con parametri metrici determinati dalla distribuzione di massa del pianeta e dalla sua rotazione.

Una mappa cartografica, d'altra parte, è un sottoinsieme aperto (o chiusura di un aperto) del piano euclideo $\mathbb{R}^2$. Il problema cartografico fondamentale consiste nel costruire una mappa:

$$\varphi: S \rightarrow \mathbb{R}^2$$

dove $S \subset M$ è una regione sulla superficie terrestre M , tale che φ sia almeno di classe C^1 (continuamente differenziabile) e soddisfi certi criteri di ottimalità rispetto alla preservazione di proprietà geometriche.

1.2 Il Theorema Egregium e le Sue Conseguenze

Il risultato centrale che governa l'intera teoria cartografica è il **Theorema Egregium** (Teorema Notevole) di Carl Friedrich Gauss, pubblicato nelle sue *Disquisitiones generales circa superficies curvas* del 1827. Questo teorema afferma che la curvatura gaussiana di una superficie è una proprietà intrinseca della geometria della superficie stessa, determinata esclusivamente dalla prima forma fondamentale (il tensore metrico), e quindi invariante sotto isometrie locali.

Teorema 1.1 (Theorema Egregium): Sia S una superficie regolare in $\mathbb{R}^3$ con prima forma fondamentale $I = E du^2 + 2F du dv + G dv^2$. La curvatura gaussiana

$$K = (LN - M^2)/(EG - F^2)$$

dove L, M, N sono i coefficienti della seconda forma fondamentale, dipende solo da E, F, G e dalle loro derivate parziali, non dall'immersione particolare della superficie in $\mathbb{R}^3$.

Dimostrazione completa: La dimostrazione procede calcolando le equazioni di Gauss (formule di compatibilità per i simboli di Christoffel) e le equazioni di Codazzi-Mainardi. Definiamo i simboli di Christoffel di prima specie:

$$\Gamma_{ij}^k = (1/2)(\partial g_{ij}/\partial u^k + \partial g_{ik}/\partial u^j - \partial g_{jk}/\partial u^i)$$

e di seconda specie:

$$\Gamma^i{}_{jk} = g^{il} \Gamma_{ljk}$$

dove g_{ij} sono le componenti del tensore metrico e g^{ij} le componenti del tensore metrico inverso. Il tensore di curvatura di Riemann è:

$$R^i{}_{jkl} = \partial \Gamma^i{}_{jk} / \partial u^l - \partial \Gamma^i{}_{lk} / \partial u^j + \Gamma^i{}_{lm} \Gamma^m{}_{jk} - \Gamma^i{}_{jm} \Gamma^m{}_{lk}$$

Per una superficie bidimensionale immersa in R^3 , usando coordinate $(u^1, u^2) = (u, v)$, la curvatura gaussiana è:

$$K = R^1{}_{212} / (g_{11}g_{22} - g_{12}^2) = R^1{}_{212} / (EG - F^2)$$

Sviluppando esplicitamente $R^1{}_{212}$ in termini dei simboli di Christoffel:

$$R^1{}_{212} = \partial \Gamma^1{}_{12} / \partial v - \partial \Gamma^1{}_{11} / \partial u + \Gamma^1{}_{1j} \Gamma^j{}_{22} - \Gamma^1{}_{2j} \Gamma^j{}_{11}$$

Sostituendo le espressioni dei $\Gamma^i{}_{jk}$ in termini di E, F, G e loro derivate, dopo algebra laboriosa otteniamo:

$$K = -1/(2\sqrt{(EG-F^2)}) [\partial/\partial u (E_v/\sqrt{(EG-F^2)}) + \partial/\partial v (G_u/\sqrt{(EG-F^2)})]$$

che dipende solo dalla prima forma fondamentale. $\square$

Corollario 1.2: Per una sfera di raggio R, la curvatura gaussiana è costante $K = 1/R^2$. Per un piano, $K = 0$ ovunque.

Corollario 1.3 (Impossibilità Fondamentale): Non esiste un'isometria locale (trasformazione che preserva lunghezze infinitesimali) tra una sfera (o ellissoide) e un piano.

Dimostrazione: Supponiamo per assurdo che esista un'isometria locale $f: S^2 \rightarrow R^2$. Allora f preserverebbe la prima forma fondamentale e quindi, per il Theorema Egregium, la curvatura gaussiana. Ma $K_{S^2} = 1/R^2 > 0$ mentre $K_{R^2} = 0$, una contraddizione. $\square$

Questo risultato stabilisce matematicamente che ogni rappresentazione cartografica deve necessariamente introdurre distorsioni. Non è possibile preservare simultaneamente tutte le proprietà metriche (lunghezze, angoli, aree) nel passaggio dalla superficie curva al piano.

1.3 Classificazione delle Proprietà Geometriche

Una trasformazione cartografica può preservare (localmente o globalmente) diverse proprietà geometriche:

Definizione 1.4 (Proiezione Conforme): Una mappa $\varphi: S \rightarrow R^2$ è conforme (o isogonale) se preserva gli angoli tra curve che si intersecano. Matematicamente, se v e w sono due vettori tangenti in un punto $p \in S$, allora:

$$\angle(d\varphi(v), d\varphi(w)) = \angle(v, w)$$

Definizione 1.5 (Proiezione Equivalente): Una mappa $\varphi: S \rightarrow \mathbb{R}^2$ è equivalente (o authalica o equal-area) se preserva le aree. Per ogni regione misurabile $D \subset S$:

$$\text{Area}(\varphi(D)) = \text{Area}(D)$$

Definizione 1.6 (Proiezione Equidistante): Una mappa $\varphi: S \rightarrow \mathbb{R}^2$ è equidistante lungo una famiglia di curve se preserva le lunghezze lungo quelle curve specifiche.

Teorema 1.7 (Incompatibilità): Non esiste una proiezione che sia simultaneamente conforme ed equivalente su una regione di misura positiva, eccetto per le isometrie locali (che non esistono tra sfera e piano per il Corollario 1.3).

Dimostrazione: Supponiamo che φ sia sia conforme che equivalente. Per conformità, il differenziale $d\varphi$ è una similitudine (rotazione + riscaldamento uniforme) con fattore di scala $\lambda(p)$. Per equivalenza, il determinante del Jacobiano deve essere 1 ovunque: $\det(J\varphi) = 1$. Ma per una similitudine, $\det(J\varphi) = \lambda^2$. Quindi $\lambda^2 = 1$, cioè $\lambda = 1$ ovunque. Questo significa che $d\varphi$ è un'isometria lineare in ogni punto, quindi φ è un'isometria locale, che non può esistere. $\square$

Capitolo 2: Geometria Differenziale dell'Ellissoide Terrestre

2.1 Derivazione Fisica dell'Ellissoide di Rotazione

La forma della Terra è determinata dall'equilibrio idrostatico di un corpo fluido rotante sotto l'azione della gravità. Consideriamo un pianeta omogeneo di densità ρ che ruota con velocità angolare costante Ω attorno a un asse. Ogni elemento di massa dm sperimenta:

1. **Forza gravitazionale:** $dF_g = -G \int (r - r')/|r - r'|^3 \rho dV'$
2. **Forza centrifuga:** $dF_c = \Omega \times (\Omega \times r) dm$

All'equilibrio, la superficie deve essere una superficie equipotenziale del potenziale efficace:

$$\Phi_{\text{eff}}(r) = \Phi_{\text{grav}}(r) + \Phi_{\text{cent}}(r)$$

Per una sfera omogenea di massa M e raggio a :

$$\Phi_{\text{grav}}(r) = -GM/r \quad (r \geq a)$$

Il potenziale centrifugo è:

$$\Phi_{\text{cent}}(r, \theta) = -(1/2)\Omega^2 r^2 \sin^2\theta = -(1/2)\Omega^2 s^2$$

dove θ è la colatitudine ($\theta = \pi/2 - \varphi$ dove φ è la latitudine) e $s = r \sin \theta$ è la distanza dall'asse di rotazione.

Il potenziale totale è:

$$\Phi_{\text{eff}}(r, \theta) = -GM/r - (1/2)\Omega^2 r^2 \sin^2\theta$$

Per piccole deformazioni rispetto alla sfera (valido per la Terra dove lo schiacciamento $f \approx 1/298$), possiamo sviluppare in serie usando i polinomi di Legendre. Scriviamo:

$$r(\theta) = a[1 + \sum \varepsilon_n P_n(\cos \theta)]$$

dove ε_n sono piccoli parametri e P_n sono i polinomi di Legendre:

$$P_0(x) = 1 \quad P_1(x) = x \quad P_2(x) = (3x^2 - 1)/2 \quad P_3(x) = (5x^3 - 3x)/2 \dots$$

L'equilibrio richiede che Φ_{eff} sia costante sulla superficie. Al primo ordine in $\Omega^2 a^3 / (GM) = m$ (rapporto tra forza centrifuga e gravitazionale all'equatore):

$$\varepsilon_2 = m/2 = \Omega^2 a^3 / (2GM)$$

Per la Terra:

- $\Omega = 7.292115 \times 10^{-5} \text{ rad/s}$
- $a = 6.378137 \times 10^6 \text{ m}$
- $M = 5.972 \times 10^{24} \text{ kg}$
- $G = 6.674 \times 10^{-11} \text{ m}^3/(\text{kg} \cdot \text{s}^2)$

Quindi:

$$m = (7.292115 \times 10^{-5})^2 \times (6.378137 \times 10^6)^3 / (2 \times 6.674 \times 10^{-11} \times 5.972 \times 10^{24}) \approx 3.46 \times 10^{-3}$$

$$\text{e } \varepsilon_2 \approx 1.73 \times 10^{-3}.$$

Questo predice uno schiacciamento:

$$f = \varepsilon_2 \approx 1/578$$

Il valore osservato $f \approx 1/298$ è circa il doppio, perché la Terra non è omogenea ma ha una densità crescente verso il centro (nucleo ferroso). La teoria completa richiede l'integrazione delle equazioni di Clairaut che includono la distribuzione di densità $\rho(r)$.

2.2 Parametrizzazione Matematica dell'Ellissoide

Un ellissoide di rotazione con semiasse equatoriale a e semiasse polare b ha equazione cartesiana:

$$x^2/a^2 + y^2/a^2 + z^2/b^2 = 1$$

Definiamo lo **schiacciamento** (flattening):

$$f = (a - b)/a$$

e l'**eccentricità** (prima e seconda):

$$e^2 = (a^2 - b^2)/a^2 = 2f - f^2 \approx 2f \quad e'^2 = (a^2 - b^2)/b^2 = e^2/(1 - e^2)$$

Per la Terra (ellissoide WGS84):

- $a = 6378137.0 \text{ m}$
- $f^{-1} = 298.257223563$
- $b = a(1 - f) = 6356752.314245 \text{ m}$
- $e^2 = 0.00669437999014$
- $e'^2 = 0.00673949674228$

Una parametrizzazione standard usa le coordinate geografiche (λ, φ) :

- λ : longitudine (angolo rispetto al meridiano di Greenwich, $-\pi \leq \lambda \leq \pi$)
- φ : latitudine geodetica (angolo che la normale alla superficie fa con il piano equatoriale, $-\pi/2 \leq \varphi \leq \pi/2$)

La parametrizzazione cartesiana è:

$$x(\lambda, \varphi) = N(\varphi) \cos \varphi \cos \lambda \quad y(\lambda, \varphi) = N(\varphi) \cos \varphi \sin \lambda \quad z(\varphi) = N(\varphi)(1 - e^2) \sin \varphi$$

dove $N(\varphi)$ è il **raggio di curvatura del primo verticale** (o grande normale):

$$N(\varphi) = a/\sqrt{1 - e^2 \sin^2 \varphi}$$

Derivazione di $N(\varphi)$: La normale alla superficie ellissoidale in un punto (x, y, z) ha direzione proporzionale a $(x/a^2, y/a^2, z/b^2)$. Per un punto sulla superficie con latitudine geodetica φ , la normale fa angolo φ con il piano equatoriale. Quindi:

$$\tan \varphi = (z/b^2)/(\sqrt{(x^2 + y^2)}/a^2) = (a^2/b^2) \cdot z/\sqrt{(x^2 + y^2)}$$

Ma dalla parametrizzazione:

$$z = N(\varphi)(1 - e^2) \sin \varphi \quad \sqrt{(x^2 + y^2)} = N(\varphi) \cos \varphi$$

Quindi:

$$\tan \varphi = (a^2/b^2) \cdot [N(\varphi)(1 - e^2) \sin \varphi]/[N(\varphi) \cos \varphi] = (a^2/b^2)(1 - e^2) \tan \varphi$$

Usando $b^2 = a^2(1 - e^2)$:

$$\tan \varphi = (a^2/[a^2(1 - e^2)])(1 - e^2) \tan \varphi = \tan \varphi \quad \checkmark$$

Questo verifica la consistenza. Per trovare N , usiamo che il punto è sulla superficie:

$$x^2/a^2 + y^2/a^2 + z^2/b^2 = 1 \quad N^2 \cos^2 \varphi/a^2 + N^2(1 - e^2)^2 \sin^2 \varphi/b^2 = 1$$

Usando $b^2 = a^2(1 - e^2)$:

$$N^2 \cos^2 \varphi/a^2 + N^2(1 - e^2) \sin^2 \varphi/a^2 = 1 \quad N^2[\cos^2 \varphi + (1 - e^2) \sin^2 \varphi]/a^2 = 1 \quad N^2 = a^2/[\cos^2 \varphi + (1 - e^2) \sin^2 \varphi]$$

Quindi:

$$N(\varphi) = a/\sqrt{1 - e^2 \sin^2 \varphi} \quad \checkmark$$

2.3 Raggi di Curvatura dell'Ellissoide

Un ellissoide ha curvature diverse in direzioni diverse. I due raggi di curvatura principali sono:

Raggio di curvatura meridiano $M(\varphi)$: curvatura nella direzione nord-sud (lungo i meridiani)

$$M(\varphi) = a(1 - e^2)/(1 - e^2 \sin^2 \varphi)^{3/2}$$

Raggio di curvatura del primo verticale $N(\varphi)$: curvatura nella direzione est-ovest (perpendicolare ai meridiani)

$$N(\varphi) = a/(1 - e^2 \sin^2 \varphi)^{1/2}$$

Derivazione di $M(\varphi)$: Il raggio di curvatura meridiano è il raggio di curvatura della curva ottenuta intersecando l'ellissoide con un piano meridiano (piano contenente l'asse di rotazione). In un piano meridiano, l'ellissoide appare come un'ellisse con equazione:

$$\rho^2/a^2 + z^2/b^2 = 1$$

dove $\rho = \sqrt{x^2 + y^2}$. Parametizziamo questa ellisse con la latitudine parametrica β :

$$\rho = a \cos \beta \quad z = b \sin \beta$$

La latitudine geodetica φ è legata a β da:

$$\tan \varphi = (b/a) \tan \beta / (\cos \beta / \cos \beta) = (b/a) \tan \beta / (\cos \beta / \cos \beta)$$

Aspetta, ripartiamo. Usiamo la formula standard per la curvatura di una curva piana $y = f(x)$:

$$\kappa = |f''|/(1 + f'^2)^{3/2}$$

Per l'ellisse meridiana parametrizzata come $z = z(\rho)$:

$$z^2 = b^2(1 - \rho^2/a^2) \quad z = b\sqrt{1 - \rho^2/a^2}$$

$$dz/d\rho = b \cdot (-2\rho/a^2)/(2\sqrt{1 - \rho^2/a^2}) = -b\rho/(a^2\sqrt{1 - \rho^2/a^2})$$

$$d^2z/d\rho^2 = -b/(a^2\sqrt{1 - \rho^2/a^2}) - b\rho \cdot [\rho/a^2] / [2(1 - \rho^2/a^2)^{3/2}] = -b/(a^2\sqrt{1 - \rho^2/a^2}) - b\rho^2/(a^4(1 - \rho^2/a^2)^{3/2}) = -b/[a^2(1 - \rho^2/a^2)^{3/2}]$$

Il raggio di curvatura è:

$$R = 1/\kappa = (1 + (dz/d\rho)^2)^{3/2}/|d^2z/d\rho^2|$$

Dopo sostituzione e semplificazione (usando $\rho = N \cos \varphi$):

$$M(\varphi) = a(1 - e^2)/(1 - e^2 \sin^2 \varphi)^{3/2} \quad \checkmark$$

Relazione tra M e N:

$$M(\varphi) = N(\varphi) \cdot (1 - e^2)/(1 - e^2 \sin^2 \varphi) = N(\varphi)(1 - e^2)^{1/2} \cdot (1 - e^2 \sin^2 \varphi)^{-1}$$

All'equatore ($\varphi = 0$):

- $M(0) = a(1 - e^2)$
- $N(0) = a$
- $M(0)/N(0) = 1 - e^2 \approx 0.9933$

Ai poli ($\varphi = \pm\pi/2$):

- $M(\pi/2) = a(1 - e^2)/(1 - e^2)^{3/2} = a/(1 - e^2)^{1/2} = a/b \cdot b = a^2/b$
- $N(\pi/2) = a/(1 - e^2)^{1/2} = a^2/b$
- $M(\pi/2) = N(\pi/2)$

La curvatura è minima all'equatore (raggio massimo) e massima ai poli.

2.4 Prima Forma Fondamentale dell'Ellissoide

La prima forma fondamentale (tensore metrico) codifica tutta la geometria intrinseca della superficie. Per l'ellissoide parametrizzato da (λ, φ) , la forma generale è:

$$ds^2 = E d\lambda^2 + 2F d\lambda d\varphi + G d\varphi^2$$

Calcoliamo i coefficienti. Il vettore posizione è:

$$r(\lambda, \varphi) = (N \cos \varphi \cos \lambda, N \cos \varphi \sin \lambda, N(1-e^2) \sin \varphi)$$

Le derivate parziali sono:

$$r_{\lambda} = \partial r / \partial \lambda = (-N \cos \varphi \sin \lambda, N \cos \varphi \cos \lambda, 0) \quad r_{\varphi} = \partial r / \partial \varphi = (\partial N / \partial \varphi \cos \varphi \cos \lambda - N \sin \varphi \cos \lambda, \partial N / \partial \varphi \cos \varphi \sin \lambda - N \sin \varphi \sin \lambda, \partial N / \partial \varphi (1-e^2) \sin \varphi + N(1-e^2) \cos \varphi)$$

Calcoliamo $\partial N / \partial \varphi$:

$$N(\varphi) = a(1 - e^2 \sin^2 \varphi)^{-1/2}$$

$$\begin{aligned} \partial N / \partial \varphi &= a \cdot (-1/2)(1 - e^2 \sin^2 \varphi)^{-3/2} \cdot (-2e^2 \sin \varphi \cos \varphi) = a e^2 \sin \varphi \cos \varphi / (1 - e^2 \sin^2 \varphi)^{3/2} \\ &= N e^2 \sin \varphi \cos \varphi / (1 - e^2 \sin^2 \varphi) = N^3 e^2 \sin \varphi \cos \varphi / a^2 \end{aligned}$$

I coefficienti della prima forma fondamentale sono:

$$E = r_{\lambda} \cdot r_{\lambda} = N^2 \cos^2 \varphi (\sin^2 \lambda + \cos^2 \lambda) = N^2 \cos^2 \varphi$$

Per G, dobbiamo calcolare $r_{\varphi} \cdot r_{\varphi}$. Questo è algebricamente laborioso. Il risultato finale è:

$$G = M^2$$

dove $M = a(1 - e^2)/(1 - e^2 \sin^2 \varphi)^{3/2}$ è il raggio di curvatura meridiano.

Il coefficiente misto è:

$$F = r_\lambda \cdot r_\varphi = 0$$

Le coordinate (λ, φ) sono **ortogonali**: le linee di λ costante (meridiani) e φ costante (paralleli) si intersecano ad angolo retto.

La prima forma fondamentale dell'ellissoide WGS84 è quindi:

$$ds^2 = (N \cos \varphi)^2 d\lambda^2 + M^2 d\varphi^2$$

2.5 Curvatura Gaussiana dell'Ellissoide

La curvatura gaussiana può essere calcolata dalla formula di Gauss per coordinate ortogonali ($F = 0$):

$$K = -1/(2\sqrt{EG}) [\partial/\partial\lambda(G_\varphi/\sqrt{EG}) + \partial/\partial\varphi(E_\lambda/\sqrt{EG})]$$

Poiché $E = N^2 \cos^2 \varphi$ e $G = M^2$ non dipendono da λ :

$$K = -1/(2NM \cos \varphi) \cdot \partial/\partial\varphi[(\partial E/\partial\varphi)/(NM \cos \varphi)]$$

Dopo calcoli (usando le espressioni di N e M):

$$K(\varphi) = 1/(MN \cos \varphi)$$

Più esplicitamente:

$$K(\varphi) = (1 - e^2)/[a^2 \cos \varphi (1 - e^2 \sin^2 \varphi)^2]$$

Questa formula mostra che:

1. $K > 0$ ovunque (curvatura positiva, come ci aspettiamo per un ellissoide)
2. K varia con la latitudine
3. All'equatore: $K(0) = (1 - e^2)/a^2$
4. Ai poli: $K(\pi/2) = 1/(a^2(1-e^2)) = 1/b^2$

Il rapporto:

$$K(\pi/2)/K(0) = 1/[(1-e^2)^2] \approx 1 + 2e^2 \approx 1.0134$$

La curvatura ai poli è circa 1.3% maggiore che all'equatore.

2.6 Tipi di Latitudine: Analisi Completa

Esistono diverse definizioni di latitudine, ognuna con sue applicazioni specifiche:

2.6.1 Latitudine Geodetica φ

È l'angolo che la normale alla superficie ellissoidale fa con il piano equatoriale. Questa è la latitudine standard usata in GPS, cartografia, e navigazione.

2.6.2 Latitudine Geocentrica φ_c

È l'angolo che il raggio vettore dal centro dell'ellissoide fa con il piano equatoriale:

$$\tan \varphi_c = z/\sqrt{(x^2 + y^2)}$$

Dalla parametrizzazione dell'ellissoide:

$$\tan \varphi_c = [N(1-e^2) \sin \varphi]/[N \cos \varphi] = (1-e^2) \tan \varphi = (b/a)^2 \tan \varphi$$

Quindi:

$$\varphi_c = \arctan[(1-e^2) \tan \varphi]$$

La differenza $\varphi - \varphi_c$ è:

$$\varphi - \varphi_c = \arctan[(1/(1-e^2) - 1) \sin \varphi \cos \varphi / (1-e^2 \sin^2 \varphi + \cos^2 \varphi)]$$

Per piccolo e^2 :

$$\varphi - \varphi_c \approx (e^2/2) \sin 2\varphi$$

Massima a $\varphi = 45^\circ$:

$$(\varphi - \varphi_c)_{\max} \approx e^2/2 \approx 0.00335 \text{ rad} \approx 11.6 \text{ arcmin}$$

Questa è una differenza significativa! Un punto con latitudine geodetica 45° ha latitudine geocentrica circa $44^\circ 48'$.

2.6.3 Latitudine Ridotta (o Parametrica) β

Definita dalla parametrizzazione della sfera ausiliaria di raggio a:

$$x = a \cos \beta \cos \lambda \quad y = a \cos \beta \sin \lambda \quad z = b \sin \beta$$

Relazione con latitudine geodetica:

$$\tan \beta = (b/a) \tan \varphi = (1-f) \tan \varphi$$

Oppure:

$$\beta = \arctan[(1-f) \tan \varphi]$$

La latitudine ridotta è fondamentale nelle trasformazioni conformi dell'ellissoide.

2.6.4 Latitudine Authalica (Equal-Area) φ_a

Definita in modo che la differenza di latitudine authalica sia proporzionale alla differenza di area sulla sfera:

$$\int_0^{\varphi} \varphi_a \cos \varphi'_a d\varphi'_a = \int_0^{\varphi} M \cos \varphi' d\varphi'/a$$

Questo porta a:

$$\sin \varphi_a = q(\varphi)/q(\pi/2)$$

dove:

$$q(\varphi) = (1-e^2)[\sin \varphi/(1-e^2 \sin^2 \varphi) - (1/(2e))\ln((1-e \sin \varphi)/(1+e \sin \varphi))]$$

2.6.5 Latitudine Conforme (Isometrica) ψ

Definita in modo che la trasformazione da (λ, φ) a (λ, ψ) sia conforme sulla sfera:

$$\psi = \ln \tan(\pi/4 + \varphi/2) - (e/2)\ln[(1 + e \sin \varphi)/(1 - e \sin \varphi)] = \ln \tan(\pi/4 + \varphi/2) - e \operatorname{arctanh}(e \sin \varphi)$$

Questa è la quantità fondamentale per la proiezione di Mercatore sull'ellissoide.

2.7 Lunghezza di Archi Geodetici

Un arco meridiano ($\lambda = \text{const}$) da φ_1 a φ_2 ha lunghezza:

$$s_m = \int_{\varphi_1}^{\varphi_2} M(\varphi) d\varphi = a(1-e^2) \int_{\varphi_1}^{\varphi_2} (1 - e^2 \sin^2 \varphi)^{-3/2} d\varphi$$

Questo integrale non ha forma chiusa elementare. Si sviluppa in serie:

$$(1 - e^2 \sin^2 \varphi)^{-3/2} = \sum_{n=0}^{\infty} C_n \sin^{2n} \varphi$$

dove i coefficienti C_n possono essere trovati dal teorema binomiale generalizzato. Usando identità trigonometriche per ridurre le potenze di $\sin \varphi$:

$$\sin^2 \varphi = (1 - \cos 2\varphi)/2 \quad \sin^4 \varphi = (3 - 4\cos 2\varphi + \cos 4\varphi)/8 \dots$$

Si ottiene:

$$s_m = a(1-e^2)[A_0\varphi - A_2 \sin 2\varphi + A_4 \sin 4\varphi - A_6 \sin 6\varphi + \dots]_{\varphi_1}^{\varphi_2}$$

dove:

$$A_0 = 1 + (3/4)e^2 + (45/64)e^4 + (175/256)e^6 + \dots \quad A_2 = (3/8)e^2 + (15/32)e^4 + (525/1024)e^6 + \dots \\ A_4 = (15/256)e^4 + (105/1024)e^6 + \dots \quad A_6 = (35/3072)e^6 + \dots$$

Per WGS84 ($e^2 \approx 0.00669$):

$$A_0 \approx 1.0050379 \quad A_2 \approx 0.0025188 \quad A_4 \approx 0.0000026 \quad A_6 \approx 0.0000000034$$

La lunghezza di un meridiano dall'equatore al polo ($\varphi = 0$ a $\varphi = \pi/2$) è:

$$s_{\text{polo}} = a(1-e^2) \cdot (\pi/2)A_0 \approx 10001965.7 \text{ m}$$

Circa 10,002 km, leggermente diversa dai 10,000 km previsti originariamente per la definizione del metro (il metro doveva essere 1/10,000,000 della distanza dal polo all'equatore).

Un arco parallelo ($\varphi = \text{const}$) da λ_1 a λ_2 ha lunghezza:

$$s_p = N(\varphi) \cos \varphi (\lambda_2 - \lambda_1)$$

All'equatore ($\varphi = 0$):

$$s_{\text{eq}} = a(\lambda_2 - \lambda_1)$$

La circonferenza equatoriale è:

$$C_{\text{eq}} = 2\pi a = 40075016.7 \text{ m}$$

Capitolo 3: Teoria Generale delle Proiezioni Cartografiche

3.1 Definizione Formale e Spazio delle Proiezioni a

Definizione 3.1: Una proiezione cartografica è un'applicazione differenziabile

$$\varphi: D \subset S \rightarrow \mathbb{R}^2$$

dove S è una superficie di riferimento (sfera o ellissoide) e $D \subset S$ è il dominio della proiezione (tipicamente $D = S \setminus \{\text{insieme finito di punti}\}$). In coordinate locali:

$$\varphi(\lambda, \theta) = (x(\lambda, \theta), y(\lambda, \theta))$$

dove λ rappresenta la longitudine e θ una qualche forma di latitudine (geodetica, geocentrica, o altra a seconda della convenzione). La scelta del dominio D è cruciale: per molte proiezioni è necessario escludere punti singolari dove la mappa non è differenziabile o dove il Jacobiano si annulla.

Lo spazio funzionale delle proiezioni cartografiche è lo spazio $C^1(D, \mathbb{R}^2)$ delle applicazioni continuamente differenziabili. In pratica, le proiezioni utilizzate sono almeno C^2 e spesso analitiche (C^ω), permettendo sviluppi in serie di potenze che facilitano sia l'analisi teorica sia l'implementazione numerica.

3.2 Il Differenziale e il Tensore di Distorsione

Il comportamento locale di una proiezione è completamente determinato dal suo differenziale. Sia $p \in S$ un punto sulla superficie di riferimento con coordinate locali (λ, θ) . Il differenziale della proiezione φ in p è l'applicazione lineare:

$$d\varphi_p: T_p S \rightarrow T_{\varphi(p)} \mathbb{R}^2$$

dove $T_p S$ è lo spazio tangente a S in p . In coordinate, $d\varphi_p$ è rappresentato dalla matrice Jacobiana:

$$J = \begin{bmatrix} \partial x / \partial \lambda & \partial x / \partial \theta \\ \partial y / \partial \lambda & \partial y / \partial \theta \end{bmatrix}$$

Questa matrice trasforma vettori tangenti sulla superficie in vettori sul piano. Le proprietà geometriche della proiezione (distorsione di lunghezze, angoli, aree) sono codificate nelle proprietà algebriche di J .

Analisi della distorsione lineare: Consideriamo un vettore tangente unitario $v \in T_p S$ con direzione α rispetto al meridiano locale. La sua immagine $w = d\varphi_p(v)$ ha lunghezza:

$$|w| = h(\alpha) = \sqrt{h^2 \cos^2 \alpha + k^2 \sin^2 \alpha + 2hk \sin \alpha \cos \alpha \sin \omega}$$

dove:

- h è il fattore di scala meridiano
- k è il fattore di scala parallelo
- ω è l'angolo di convergenza dei meridiani

Per coordinate ortogonali sulla superficie (come λ, φ per l'ellissoide), l'espressione si semplifica e $\omega = 0$.

Definizione 3.2 (Indicatrice di Tissot): L'indicatrice di Tissot (o ellisse di distorsione) in un punto p è l'immagine tramite $d\varphi_p$ di un cerchio infinitesimo unitario centrato in p sulla superficie. Geometricamente, è un'ellisse nel piano cartografico con semiassi:

$$a = (h + k)/2 + \sqrt{[(h - k)^2/4 + (hk \sin \omega)^2]} \quad b = (h + k)/2 - \sqrt{[(h - k)^2/4 + (hk \sin \omega)^2]}$$

dove $a \geq b > 0$ sono i semiassi maggiore e minore.

Teorema 3.3: Una proiezione è conforme in p se e solo se l'indicatrice di Tissot in p è un cerchio, cioè $a = b$. Una proiezione è equivalente in p se e solo se l'area dell'indicatrice è $\pi ab = \pi$.

Dimostrazione: La conformità equivale a richiedere che $d\varphi_p$ sia una similitudine (rotazione + riscaldamento uniforme), cioè che tutti i vettori unitari vengano scalati dello stesso fattore indipendentemente dalla direzione. Questo si verifica se e solo se $a = b$.

Per l'equivalenza, l'area di una regione infinitesima dA sulla superficie viene trasformata in un'area sul piano pari a $|\det(J)|dA$. L'area dell'indicatrice di Tissot è $\pi ab = \pi |\det(J)|$. Richiedendo che le aree siano preservate, dobbiamo avere $|\det(J)| = 1$, quindi $ab = 1$. $\square$

3.3 Classificazione Analitica delle Proiezioni

3.3.1 Proiezioni Conformi: Teoria di Cauchy-Riemann

Una proiezione conforme deve preservare gli angoli tra tutte le coppie di curve intersecanti. Questa è una condizione estremamente restrittiva che porta a una struttura matematica profonda.

Teorema 3.4 (Caratterizzazione delle Proiezioni Conformi): Sia $\varphi: S \rightarrow \mathbb{R}^2$ una proiezione con componenti $x(\lambda, \theta)$, $y(\lambda, \theta)$. La proiezione è conforme se e solo se le funzioni complesse:

$$z(\lambda, \theta) = x(\lambda, \theta) + iy(\lambda, \theta)$$

soddisfano le equazioni generalizzate di Cauchy-Riemann rispetto alla metrica intrinseca di S .

Dimostrazione: Scriviamo la prima forma fondamentale di S come:

$$ds^2 = E d\lambda^2 + 2F d\lambda d\theta + G d\theta^2$$

La condizione di conformità richiede che:

$$(\partial x / \partial \lambda)^2 + (\partial y / \partial \lambda)^2 = \mu^2 E \quad (\partial x / \partial \theta)^2 + (\partial y / \partial \theta)^2 = \mu^2 G \quad (\partial x / \partial \lambda)(\partial x / \partial \theta) + (\partial y / \partial \lambda)(\partial y / \partial \theta) = \mu^2 F$$

per qualche funzione $\mu(\lambda, \theta) > 0$ (il fattore di scala locale). Queste tre equazioni si possono riscrivere in forma vettoriale come:

$$\nabla x \cdot \nabla x = \mu^2 E, \quad \nabla y \cdot \nabla y = \mu^2 G, \quad \nabla x \cdot \nabla y = \mu^2 F$$

dove il gradiente è preso rispetto alla metrica della superficie. La terza equazione dice che le curve di livello di x e y sono ortogonali. Le prime due dicono che hanno la stessa "densità di livello".

Per una superficie con coordinate ortogonali ($F = 0$), come l'ellissoide in coordinate geografiche, le equazioni diventano:

$$E(\partial x / \partial \lambda)^2 + G(\partial x / \partial \theta)^2 = \mu^2 E G \quad E(\partial y / \partial \lambda)^2 + G(\partial y / \partial \theta)^2 = \mu^2 E G \quad E(\partial x / \partial \lambda)(\partial y / \partial \lambda) + G(\partial x / \partial \theta)(\partial y / \partial \theta) = 0$$

Dividendo per EG e introducendo coordinate isometriche u, v tali che $ds^2 = \lambda^2(du^2 + dv^2)$, queste si riducono alle classiche equazioni di Cauchy-Riemann:

$$\partial x / \partial u = \partial y / \partial v, \quad \partial x / \partial v = -\partial y / \partial u \quad \square$$

Corollario 3.5: In coordinate isometriche, ogni proiezione conforme corrisponde a una funzione olomorfa (analitica complessa).

Questo risultato profondo connette la teoria delle proiezioni conformi alla teoria delle funzioni di variabile complessa. L'esistenza di proiezioni conformi tra regioni di superfici curve è garantita dal teorema di uniformizzazione di Riemann, uno dei risultati centrali dell'analisi complessa e della geometria differenziale.

3.3.2 Coordinate Isometriche per l'Ellissoide

Per applicare la teoria complessa alle proiezioni conformi dell'ellissoide, dobbiamo prima costruire coordinate isometriche. Partiamo dalla prima forma fondamentale:

$$ds^2 = (N \cos \varphi)^2 d\lambda^2 + M^2 d\varphi^2$$

Cerchiamo una trasformazione di coordinate $(\lambda, \varphi) \rightarrow (\lambda, \psi)$ tale che:

$$ds^2 = f^2(\lambda, \psi)(d\lambda^2 + d\psi^2)$$

per qualche funzione f . Poiché λ rimane invariata, dobbiamo avere:

$$(N \cos \varphi)^2 = f^2 M^2 (d\varphi/d\psi)^2 = f^2$$

Da queste:

$$d\psi/d\varphi = M/(N \cos \varphi)$$

Integrando:

$$\psi(\varphi) = \int_0^\varphi M(\varphi')/(N(\varphi') \cos \varphi') d\varphi'$$

Sostituendo le espressioni di M e N :

$$M/(N \cos \varphi) = [a(1-e^2)/(1-e^2 \sin^2 \varphi)^{3/2}] / [a \cos \varphi/(1-e^2 \sin^2 \varphi)^{1/2}] = (1-e^2) / [\cos \varphi(1-e^2 \sin^2 \varphi)]$$

Questo integrale è risolvibile in forma chiusa:

$$\psi = \ln \tan(\pi/4 + \varphi/2) - (e/2) \ln[(1 + e \sin \varphi)/(1 - e \sin \varphi)]$$

Questa è precisamente la latitudine conforme (o isometrica) definita precedentemente. Possiamo anche scriverla come:

$$\psi = \operatorname{arctanh}(\sin \varphi) - e \operatorname{arctanh}(e \sin \varphi) = \ln[(1 + \sin \varphi)/(1 - \sin \varphi)]^{1/2} - (e/2) \ln[(1 + e \sin \varphi)/(1 - e \sin \varphi)]$$

In coordinate (λ, ψ) , la prima forma fondamentale diventa:

$$ds^2 = (N \cos \varphi)^2(d\lambda^2 + d\psi^2)$$

dove φ deve essere inteso come funzione di ψ (invertendo la relazione precedente, che richiede metodi numerici o sviluppi in serie).

3.3.3 Proiezioni Equivalenti: Vincoli Differenziali

Le proiezioni equivalenti devono preservare le aree. Matematicamente:

Teorema 3.6: Una proiezione $\varphi: S \rightarrow R^2$ è equivalente se e solo se:

$$\det(J\varphi) = \sqrt{(EG - F^2)}$$

ovunque, dove E, F, G sono i coefficienti della prima forma fondamentale di S.

Dimostrazione: Un elemento di area sulla superficie è:

$$dA_S = \sqrt{(EG - F^2)} d\lambda d\theta$$

L'immagine di questo elemento nel piano ha area:

$$dA_{R^2} = |\det(J\varphi)| d\lambda d\theta$$

La condizione di equivalenza $dA_{R^2} = dA_S$ implica immediatamente:

$$|\det(J\varphi)| = \sqrt{(EG - F^2)}$$

Possiamo scegliere l'orientazione in modo che $\det(J\varphi) > 0$. $\square$

Esplicitamente, per l'ellissoide:

$$\partial x / \partial \lambda \cdot \partial y / \partial \varphi - \partial x / \partial \varphi \cdot \partial y / \partial \lambda = MN \cos \varphi$$

Questa è un'equazione differenziale alle derivate parziali del primo ordine per le funzioni incognite $x(\lambda, \varphi)$ e $y(\lambda, \varphi)$. A differenza delle equazioni per proiezioni conformi (che sono ellittiche e fortemente vincolanti), questa singola equazione scalare lascia grande libertà nella scelta della proiezione equivalente.

Parametrizzazione generale: Le proiezioni equivalenti possono essere parametrizzate introducendo una funzione ausiliaria arbitraria $\alpha(\lambda, \varphi)$:

$$\begin{aligned} \partial x / \partial \lambda &= g(\lambda, \varphi) \cos \alpha & \partial y / \partial \lambda &= g(\lambda, \varphi) \sin \alpha & \partial x / \partial \varphi &= h(\lambda, \varphi) \cos(\alpha + \pi/2) = -h(\lambda, \varphi) \sin \alpha \\ \partial y / \partial \varphi &= h(\lambda, \varphi) \sin(\alpha + \pi/2) = h(\lambda, \varphi) \cos \alpha \end{aligned}$$

dove α rappresenta l'angolo che i meridiani trasformati fanno con le linee $y = \text{const}$ nel piano cartografico. La condizione di equivalenza diventa:

$$g \cdot h = MN \cos \varphi$$

Possiamo scegliere liberamente g (o equivalentemente $h = MN \cos \varphi / g$), e questo determina completamente la proiezione equivalente a meno di una traslazione e rotazione rigida nel piano.

3.4 Metriche di Distorsione

Per confrontare quantitativamente le distorsioni introdotte da diverse proiezioni, introduciamo diverse metriche standard.

3.4.1 Fattori di Scala Principale

Definiamo:

- h: fattore di scala lungo i meridiani (direzione φ)
- k: fattore di scala lungo i paralleli (direzione λ)

Per l'ellissoide in coordinate geografiche:

$$h = (1/M) |\partial r / \partial \varphi| = \sqrt{[(\partial x / \partial \varphi)^2 + (\partial y / \partial \varphi)^2] / M} \quad k = (1/(N \cos \varphi)) |\partial r / \partial \lambda| = \sqrt{[(\partial x / \partial \lambda)^2 + (\partial y / \partial \lambda)^2] / (N \cos \varphi)}$$

Proprietà fondamentali:

1. Per proiezioni conformi: $h = k$ ovunque
2. Per proiezioni equivalenti: $h \cdot k = 1$ ovunque
3. Il fattore di scala in direzione α rispetto al meridiano è:

$$s(\alpha) = \sqrt{(h^2 \cos^2 \alpha + k^2 \sin^2 \alpha)} \quad (\text{per proiezioni ortogonali})$$

3.4.2 Distorsione di Aree

Il rapporto di distorsione delle aree è:

$$\sigma = hk$$

Per proiezioni equivalenti, $\sigma = 1$ ovunque. Per proiezioni conformi, $\sigma = h^2 = k^2$.

3.4.3 Distorsione Angolare Massima

Per proiezioni non conformi, gli angoli non sono preservati. La massima distorsione angolare in un punto è:

$$\omega_{\max} = 2 \arcsin |(h - k)/(h + k)|$$

Questa quantità misura quanto un angolo retto viene distorto nel peggiore dei casi.

3.4.4 Indice di Airy

Nicolas Airy (1861) propose un indice di distorsione complessivo che bilancia tutte le direzioni:

$$B = (h^2 + k^2)/2$$

Per una proiezione conforme con fattore di scala μ , $B = \mu^2$. Per valutare la distorsione complessiva su una regione D, si usa:

$$B_{\text{medio}} = (1/\text{Area}(D)) \iint_D B \, dA$$

Una proiezione ottimale nel senso di Airy minimizza B_{medio} sulla regione di interesse.

3.5 Il Teorema di Eulero sulle Indicatrici

Teorema 3.7 (Eulero, 1777): Per ogni proiezione, esistono in ogni punto due direzioni ortogonali (direzioni principali) lungo le quali i fattori di scala raggiungono i loro valori estremi a e b (con $a \geq b$). In qualsiasi altra direzione, il fattore di scala s soddisfa $b \leq s \leq a$.

Dimostrazione: Il fattore di scala in direzione α è dato da $s^2(\alpha)$ come funzione quadratica di $\cos \alpha$ e $\sin \alpha$. Questa funzione raggiunge necessariamente un massimo e un minimo su $[0, 2\pi)$. Per trovare gli estremi, deriviamo rispetto a α e poniamo uguale a zero. Gli estremi si verificano alle direzioni principali, che per proiezioni con meridiani e paralleli ortogonali nel piano coincidono con le direzioni dei meridiani e paralleli stessi. $\square$

Corollario 3.8: Per proiezioni con meridiani e paralleli ortogonali nel piano cartografico, le direzioni principali sono proprio le direzioni dei meridiani e dei paralleli, con fattori di scala h e k .

3.6 Classificazione per Superficie Ausiliaria

Storicamente, molte proiezioni sono state concepite geometricamente come proiezioni tramite una superficie intermedia sviluppabile (superficie con curvatura gaussiana nulla che può essere "srotolata" sul piano senza distorsione).

3.6.1 Proiezioni Cilindriche

La superficie ausiliaria è un cilindro tangente o secante alla sfera/ellissoide. Il cilindro può poi essere "srotolato" su un piano.

Forma generale:

$$x = F(\lambda, \varphi)$$

$$y = G(\varphi)$$

dove λ appare linearmente in x (i meridiani diventano linee verticali parallele).

Proprietà generali:

- I meridiani sono mappati in linee rette parallele
- I paralleli sono mappati in linee rette perpendicolari ai meridiani
- La scala lungo i meridiani può variare con la latitudine
- La scala lungo i paralleli può variare con la latitudine

Casi speciali:

1. Cilindrica equidistante: $y = \varphi$, distanze preservate lungo meridiani
2. Cilindrica equivalente: y scelta per preservare aree
3. Cilindrica conforme (Mercatore): y scelto per preservare angoli

3.6.2 Proiezioni Coniche

La superficie ausiliaria è un cono tangente o secante. Il cono viene poi tagliato lungo una generatrice e srotolato.

Forma generale in coordinate polari nel piano:

$$\rho = \rho(\varphi)$$

$$\theta = n \cdot \lambda$$

dove n è la costante conica (rapporto tra l'angolo al vertice del cono srotolato e l'angolo di longitudine).

Proprietà generali:

- I meridiani diventano raggi che si dipartono da un punto (l'immagine del polo)
- I paralleli diventano archi di cerchio concentrici
- Il fattore n determina quanta "convergenza" hanno i meridiani

Casi speciali:

1. Conica conforme di Lambert: n e $\rho(\varphi)$ scelti per conformità
2. Conica equivalente di Albers: n e $\rho(\varphi)$ scelti per equivalenza
3. Conica equidistante: distanze preservate lungo meridiani

3.6.3 Proiezioni Azimutali (Planari)

La superficie ausiliaria è un piano tangente alla sfera/ellissoide in un punto (tipicamente un polo o un punto arbitrario).

Forma generale in coordinate polari centrate nel punto di tangenza:

$$\rho = \rho(c)$$

$$\theta = Az$$

dove c è la distanza angolare dal centro e Az è l'azimut.

Proprietà generali:

- I cerchi massimi passanti per il centro diventano linee rette
- Le direzioni dal centro sono preservate (da cui "azimutale")
- Simmetria radiale rispetto al centro

Casi speciali:

1. Gnomonica: $\rho = R \tan c$
2. Stereografica: $\rho = 2R \tan(c/2)$ [conforme]
3. Orografica: $\rho = R \sin c$
4. Azimutale equivalente di Lambert: $\rho = 2R \sin(c/2)$

5. Azimutale equidistante: $\rho = R\phi$

3.7 Trasformazioni tra Proiezioni

Date due proiezioni ϕ_1 e ϕ_2 della stessa superficie, possiamo comporre:

$$\phi_2 = T \circ \phi_1$$

dove $T = \phi_2 \circ \phi_1^{-1}: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ è una trasformazione del piano. Questa composizione è particolarmente utile quando:

1. ϕ_1 ha espressione analitica semplice
2. T è una trasformazione geometrica elementare (rotazione, riflessione, proiezione)
3. Il risultato ϕ_2 ha proprietà desiderate

Esempio: Le proiezioni azimutali centrate in un punto arbitrario possono essere costruite componendo:

1. Una rotazione della sfera che porta il punto al polo nord
2. Una proiezione azimutale polare standard
3. Eventualmente una rotazione nel piano immagine

La matematica di queste composizioni coinvolge algebra delle rotazioni (gruppo $SO(3)$) e geometria sferica.

Capitolo 4: Proiezioni Conformi: Teoria Completa

4.1 La Proiezione di Mercatore: Derivazione Rigorosa

La proiezione di Mercatore, introdotta da Gerardus Mercator nel 1569, è forse la più famosa proiezione cartografica. La sua importanza deriva dalla proprietà conforme, che la rende essenziale per la navigazione marittima e aerea.

4.1.1 Mercatore sulla Sfera

Partiamo dal caso sferico prima di trattare l'ellissoide. Per una sfera di raggio R , la prima forma fondamentale è:

$$ds^2 = R^2 \cos^2\phi \, d\lambda^2 + R^2 \, d\phi^2$$

Per ottenere una proiezione conforme cilindrica, cerchiamo una trasformazione della forma:

$$x = f(\lambda) \quad y = g(\phi)$$

Il requisito di conformità impone che il rapporto tra i fattori di scala meridiano e parallelo sia unitario:

$$h/k = 1$$

Calcoliamo i fattori di scala:

$$k = |\partial x / \partial \lambda| / (R \cos \varphi) = |f'(\lambda)| / (R \cos \varphi) \quad h = |\partial y / \partial \varphi| / R = |g'(\varphi)| / R$$

La condizione $h = k$ diventa:

$$|g'(\varphi)| / R = |f'(\lambda)| / (R \cos \varphi)$$

Poiché il lato sinistro dipende solo da φ e il destro solo da λ (eccetto per $\cos \varphi$), entrambi devono essere costanti. Scegliendo la normalizzazione $f(\lambda) = R\lambda$ (i meridiani sono equidistanti sull'asse x), otteniamo:

$$f'(\lambda) = R$$

Quindi:

$$g'(\varphi) = R / \cos \varphi = R \sec \varphi$$

Integrando:

$$g(\varphi) = R \int \sec \varphi \, d\varphi = R \ln |\sec \varphi + \tan \varphi| + C = R \ln |\tan(\pi/4 + \varphi/2)| + C$$

Scegliendo $C = 0$ e ponendo l'equatore come origine:

Proiezione di Mercatore (sfera):

$$x = R\lambda$$

$$y = R \ln \tan(\pi/4 + \varphi/2)$$

Proprietà fondamentali:

1. **Conformità:** Per costruzione, preserva angoli localmente
2. **Lossodromie come linee rette:** Una lossodromia (linea di azimuth costante) sulla sfera appare come una retta nel piano di Mercatore
3. **Distorsione polare infinita:** Quando $\varphi \rightarrow \pm\pi/2$, $y \rightarrow \pm\infty$. I poli non possono essere rappresentati
4. **Fattore di scala:** $\mu(\varphi) = \sec \varphi$

La distorsione areale è:

$$\sigma = \mu^2 = \sec^2 \varphi$$

All'equatore: $\sigma = 1$ (nessuna distorsione) A $\varphi = 60^\circ$: $\sigma = 4$ (aree raddoppiate in ogni dimensione, quadruplicate in area) A $\varphi = 80^\circ$: $\sigma \approx 33.6$

4.1.2 Mercatore sull'Ellissoide

Per l'ellissoide WGS84, dobbiamo usare la latitudine conforme ψ definita precedentemente. La proiezione diventa:

Proiezione di Mercatore (ellissoide):

$$x = a\lambda$$

$$y = a \cdot \psi(\varphi)$$

dove:

$$\psi(\varphi) = \ln \tan(\pi/4 + \varphi/2) - (e/2) \ln[(1 + e \sin \varphi)/(1 - e \sin \varphi)]$$

Questa può anche essere scritta come:

$$\psi = \operatorname{arctanh}(\sin \varphi) - e \cdot \operatorname{arctanh}(e \sin \varphi)$$

oppure, usando la funzione di Gudermann inversa:

$$\psi = \operatorname{gd}^{-1}(\varphi) - e \cdot \operatorname{arctanh}(e \sin \varphi)$$

Inversione: Data una coppia (x, y) nel piano di Mercatore, trovare (λ, φ) richiede:

$$\lambda = x/a \text{ (immediato)}$$

$$\varphi = \varphi(\psi) \text{ dove } \psi = y/a$$

L'inversione $\varphi = \varphi(\psi)$ non ha forma chiusa e richiede iterazione. Il metodo di Newton converge rapidamente:

$$\varphi_{i+1} = \varphi_i - [\psi - \psi(\varphi_i)]/[d\psi/d\varphi]_{\varphi_i}$$

dove:

$$d\psi/d\varphi = (1 - e^2)/(\cos \varphi \cdot (1 - e^2 \sin^2 \varphi))$$

Inizializzando con $\varphi_0 = 2 \arctan(\exp(\psi)) - \pi/2$ (la soluzione sferica), 3-4 iterazioni danno precisione a livello di millimetri.

4.1.3 Mercatore Trasversa (UTM)

La proiezione Mercatore Trasversa è ottenuta ruotando il cilindro di 90° in modo che sia tangente lungo un meridiano invece che lungo l'equatore. Questa è la base del sistema di coordinate Universal Transverse Mercator (UTM).

Costruzione geometrica:

1. Ruotiamo la sfera/ellissoide in modo che il meridiano di interesse diventi l'equatore di un nuovo sistema di coordinate
2. Applichiamo la proiezione di Mercatore standard

3. Ruotiamo il risultato nel piano

Forma analitica per l'ellissoide: Le formule sono considerevolmente più complesse. Definiamo:

- λ_0 : meridiano centrale
- φ_0 : latitudine di origine (spesso 0°)
- k_0 : fattore di scala al meridiano centrale (tipicamente 0.9996 per UTM)

Le coordinate sono espresse come serie di potenze:

$$x = k_0 N [A + (1 - T + C)A^3/6 + (5 - 18T + T^2 + 72C - 58e'^2)A^5/120 + \dots]$$

$$y = k_0 \{M - M_0 + N \tan \varphi [A^2/2 + (5 - T + 9C + 4C^2)A^4/24 + (61 - 58T + T^2 + 600C - 330e'^2)A^6/720 + \dots]\}$$

dove:

$$A = (\lambda - \lambda_0) \cos \varphi \quad T = \tan^2 \varphi \quad C = e'^2 \cos^2 \varphi \quad M = a[(1 - e^2/4 - 3e^4/64 - 5e^6/256 - \dots)\varphi - (3e^2/8 + 3e^4/32 + 45e^6/1024 + \dots) \sin 2\varphi + (15e^4/256 + 45e^6/1024 + \dots) \sin 4\varphi - (35e^6/3072 + \dots) \sin 6\varphi + \dots]$$

M_0 è il valore di M alla latitudine di origine.

Sistema UTM:

- Il mondo è diviso in 60 zone di 6° di longitudine ciascuna
- Ogni zona usa un meridiano centrale
- False easting: 500,000 m (per evitare coordinate negative)
- False northing: 0 m nell'emisfero nord, 10,000,000 m nell'emisfero sud
- Validità: da 80°S a 84°N

Precisione: All'interno di una zona UTM ($\pm 3^\circ$ dal meridiano centrale), la distorsione massima è circa:

- Al meridiano centrale: $k = 0.9996$ (0.04% di riduzione)
- Ai bordi della zona ($\pm 3^\circ$): $k \approx 1.0004$ (0.04% di allungamento)
- Distorsione massima: circa 0.08%

4.2 La Proiezione Stereografica

La proiezione stereografica è una proiezione azimutale conforme di importanza teorica e pratica straordinaria.

4.2.1 Definizione Geometrica

Consideriamo una sfera di raggio R . Il punto di proiezione è un polo (diciamo il polo sud S). Il piano di proiezione è tangente al polo opposto (polo nord N).

Per un punto P sulla sfera, la proiezione stereografica P' è l'intersezione della retta SP con il piano tangente in N .

Derivazione analitica: Usiamo coordinate sferiche con origine al centro della sfera:

- Punto generico sulla sfera: $P = (R \cos \varphi \cos \lambda, R \cos \varphi \sin \lambda, R \sin \varphi)$
- Polo sud: $S = (0, 0, -R)$
- Piano tangente in N : $z = R$

La retta da S a P è parametrizzata da:

$$r(t) = S + t(P - S) = (0, 0, -R) + t(R \cos \varphi \cos \lambda, R \cos \varphi \sin \lambda, R \sin \varphi + R)$$

Imponendo $z(t) = R$:

$$-R + t(R \sin \varphi + R) = R \quad t = 2R / (R(1 + \sin \varphi)) = 2 / (1 + \sin \varphi)$$

Le coordinate x, y nel piano tangente sono:

$$x = r_x(t) = 2R \cos \varphi \cos \lambda / (1 + \sin \varphi) \quad y = r_y(t) = 2R \cos \varphi \sin \lambda / (1 + \sin \varphi)$$

Usando l'identità trigonometrica:

$$1 + \sin \varphi = 2 \sin^2(\pi/4 + \varphi/2) \cdot \csc^2(\pi/4 + \varphi/2) = 2 \cos^2(\pi/4 - \varphi/2) / \cos^2(\pi/4 + \varphi/2)$$

Dopo semplificazione:

Proiezione Stereografica Polare:

$$x = 2R \tan(\pi/4 - \varphi/2) \cos \lambda$$

$$y = 2R \tan(\pi/4 - \varphi/2) \sin \lambda$$

O in coordinate polari nel piano:

$$\rho = 2R \tan(\pi/4 - \varphi/2) = 2R \cot(\pi/4 + \varphi/2)$$

$$\theta = \lambda$$

4.2.2 Dimostrazione della Conformità

Teorema 4.1: La proiezione stereografica è conforme.

Dimostrazione geometrica: Dobbiamo dimostrare che l'angolo tra due curve sulla sfera è preservato nel piano. Sia P un punto sulla sfera con colatitudine $\theta = \pi/2 - \varphi$. Consideriamo due curve γ_1 e γ_2 passanti per P con vettori tangenti v_1 e v_2 .

L'applicazione stereografica è un'inversione composta con una riflessione. È un fatto noto della geometria che le inversioni preservano gli angoli (sono trasformazioni conformi). Più precisamente:

1. L'inversione rispetto a una sfera di raggio r centrata in S trasforma P in P^* con $|SP| \cdot |SP^*| = r^2$
2. Le inversioni mappano cerchi in cerchi (o linee) e preservano angoli
3. La proiezione stereografica è esattamente una tale inversione seguita da una riflessione

Dimostrazione analitica: Calcoliamo il differenziale. I fattori di scala sono:

$$h = (1/R)|\partial r/\partial \varphi| = |\partial \rho/\partial \varphi|/R$$

Calcolando:

$$\partial \rho/\partial \varphi = 2R \cdot \sec^2(\pi/4 + \varphi/2) \cdot (1/2) = R \sec^2(\pi/4 + \varphi/2)$$

Ma usando $\tan(\pi/4 - \varphi/2) = \cot(\pi/4 + \varphi/2)$:

$$\sec^2(\pi/4 + \varphi/2) = 1 + \tan^2(\pi/4 + \varphi/2) = 1 + \cot^2(\pi/4 - \varphi/2) = \csc^2(\pi/4 - \varphi/2)$$

E:

$$k = |\partial \rho/\partial \lambda|/(R \cos \varphi) = \rho/(R \cos \varphi) = 2R \tan(\pi/4 - \varphi/2)/(R \cos \varphi)$$

Usando identità trigonometriche:

$$\tan(\pi/4 - \varphi/2) = (1 - \tan(\varphi/2))/(1 + \tan(\varphi/2))$$

e dopo algebra si verifica che $h = k$, quindi la proiezione è conforme. $\square$

Fattore di scala:

$$\mu(\varphi) = 2/(1 + \sin \varphi) = \sec^2(\pi/4 + \varphi/2)$$

All'equatore ($\varphi = 0$): $\mu = 2/\sqrt{2} = \sqrt{2} \approx 1.414$ Al polo nord ($\varphi = \pi/2$): $\mu = 1$ Al polo sud ($\varphi = -\pi/2$): $\mu \rightarrow \infty$

4.2.3 Proprietà Notevoli della Stereografica

1. **Cerchi su cerchi:** La proiezione stereografica mappa cerchi sulla sfera in cerchi (o linee rette) nel piano. Questa proprietà è fondamentale in cristallografia e in teoria delle funzioni complesse.
2. **Proiezione conforme del piano complesso esteso:** La sfera di Riemann (S^2 identificata con $\mathbb{C} \cup \{\infty\}$) sotto proiezione stereografica fornisce la compattificazione standard del piano complesso.

3. **Gruppo di trasformazioni:** Le rotazioni della sfera corrispondono a trasformazioni di Möbius (trasformazioni lineari frazionarie) del piano complesso:

$$w \rightarrow (aw + b)/(cw + d) \text{ con } ad - bc = 1$$

4. **Applicazioni:** Navigazione aerea polare, mappe delle calotte polari, proiezione del cielo stellare (planisferi).

4.3 Proiezione Conforme di Lambert (Conica)

La proiezione conforme conica di Lambert è ottimale per regioni che si estendono principalmente in direzione est-ovest a latitudini medie.

4.3.1 Derivazione sulla Sfera

Cerchiamo una proiezione conica conforme con due paralleli standard (latitudini φ_1 e φ_2 dove la scala è esatta).

Forma generale conica:

$$\begin{aligned}\rho &= \rho(\varphi) \\ \theta &= n\lambda\end{aligned}$$

dove n è la costante conica da determinare.

Le coordinate cartesiane sono:

$$x = \rho \sin \theta = \rho(\varphi) \sin(n\lambda) \quad y = \rho_0 - \rho \cos \theta = \rho_0 - \rho(\varphi) \cos(n\lambda)$$

dove ρ_0 è una costante determinata dalla latitudine di origine.

I fattori di scala sono:

$$k = (\rho n)/(R \cos \varphi) \text{ (direzione parallelo)} \quad h = |d\rho/d\varphi|/R \text{ (direzione meridiano)}$$

La condizione di conformità $h = k$ implica:

$$|d\rho/d\varphi|/R = (\rho n)/(R \cos \varphi)$$

Quindi:

$$d\rho/\rho = n d\varphi/\cos \varphi = n d(\ln \tan(\pi/4 + \varphi/2))$$

Integrando:

$$\ln \rho = n \ln \tan(\pi/4 + \varphi/2) + C$$

$$\rho = C_0 \tan^n(\pi/4 + \varphi/2)$$

Per determinare n e C_0 , imponiamo che la scala sia esatta ai paralleli standard φ_1 e φ_2 :

$$k(\varphi_1) = 1 \Rightarrow \rho(\varphi_1)n = R \cos \varphi_1 \quad k(\varphi_2) = 1 \Rightarrow \rho(\varphi_2)n = R \cos \varphi_2$$

Dividendo:

$$[\rho(\varphi_1)/\rho(\varphi_2)] = [\tan^n(\pi/4 + \varphi_1/2)/\tan^n(\pi/4 + \varphi_2/2)] = \cos \varphi_1/\cos \varphi_2$$

Quindi:

$$n = \ln(\cos \varphi_1/\cos \varphi_2)/\ln[\tan(\pi/4 + \varphi_1/2)/\tan(\pi/4 + \varphi_2/2)]$$

Questa formula può essere semplificata usando la notazione:

$$t_1 = \tan(\pi/4 + \varphi_1/2), \quad t_2 = \tan(\pi/4 + \varphi_2/2)$$

$$n = \ln(\cos \varphi_1/\cos \varphi_2)/\ln(t_1/t_2)$$

Una volta trovato n :

$$C_0 = (R \cos \varphi_1)/(n \tan^n(\pi/4 + \varphi_1/2))$$

Caso particolare: Se $\varphi_1 = \varphi_2 = \varphi_0$ (proiezione con un solo parallelo standard), usando L'Hôpital:

$$n = \sin \varphi_0$$

Questo è il caso della proiezione conica conforme tangente.

4.3.2 Estensione all'Ellissoide

Per l'ellissoide, sostituiamo:

- φ con la latitudine geodetica
- $\tan(\pi/4 + \varphi/2)$ con $\exp(\psi)$ dove ψ è la latitudine conforme
- $\cos \varphi$ con appropriati fattori che includono l'eccentricità

La formula per n diventa:

$$n = \ln(N_1 \cos \varphi_1/(N_2 \cos \varphi_2))/\ln[\tan(\pi/4 + \varphi_1/2) \cdot ((1-e \sin \varphi_1)/(1+e \sin \varphi_1))^{(e/2)} / \tan(\pi/4 + \varphi_2/2) \cdot ((1-e \sin \varphi_2)/(1+e \sin \varphi_2))^{(e/2)}]$$

E la funzione ρ :

$$\rho(\varphi) = F/\tan^n(\pi/4 + \varphi/2) \cdot [(1+e \sin \varphi)/(1-e \sin \varphi)]^{(en/2)}$$

dove F è una costante di normalizzazione.

4.4 Teorema di Gauss-Lagrange sulla Conformità Minimale

Un risultato profondo della teoria delle proiezioni conformi riguarda la distorsione minima.

Teorema 4.2 (Gauss-Lagrange): Tra tutte le proiezioni conformi di una regione sferica nel piano, quella che minimizza l'integrale:

$$I = \iint_D (\mu^2 - 1) dA$$

(dove μ è il fattore di scala e D è la regione) è unica ed è caratterizzata dall'equazione differenziale:

$$\nabla^2(\ln \mu) = 0$$

cioè il logaritmo del fattore di scala è una funzione armonica.

Dimostrazione (schema): Usiamo il calcolo delle variazioni. Sia φ una proiezione conforme con fattore di scala μ , e consideriamo una variazione conforme infinitesima $\varphi + \varepsilon \delta\varphi$ con fattore di scala $\mu + \varepsilon \delta\mu$.

La variazione dell'integrale è:

$$\delta I = 2 \iint_D \mu \delta\mu dA$$

Per una variazione conforme, possiamo esprimere $\delta\mu$ in termini di una funzione scalare δu :

$$\delta\mu = \mu(\partial\delta u/\partial x + i\partial\delta u/\partial y) \text{ in coordinate complesse}$$

Usando la formula di Green e imponendo $\delta I = 0$ per variazioni arbitrarie con $\delta u = 0$ sul bordo, otteniamo:

$$\nabla^2\mu = 0 \text{ nel dominio conformemente equivalente}$$

Che in forma logaritmica diventa $\nabla^2(\ln \mu) = 0$. $\square$

Applicazione: Per progettare una proiezione conforme ottimale per una regione specifica (ad esempio, uno stato o una nazione), si risolve l'equazione di Laplace per $\ln \mu$ con appropriate condizioni al contorno. Questo porta a proiezioni conformi "a doppia proiezione" usate in alcuni sistemi cartografici nazionali.

Capitolo 5: Proiezioni Equivalenti (Equal-Area)

5.1 Teoria Generale delle Proiezioni Equivalenti

Le proiezioni equivalenti preservano le aree ma necessariamente distorcono angoli e forme. Sono essenziali per rappresentazioni tematiche, densità di popolazione, distribuzione di risorse, e ogni applicazione dove l'accuratezza delle aree è prioritaria.

5.1.1 Il Vincolo di Equivalenza come EDP

La condizione di equivalenza per una proiezione $x = x(\lambda, \varphi)$, $y = y(\lambda, \varphi)$ è:

$$\partial(x,y)/\partial(\lambda,\varphi) = \sqrt{(EG - F^2)}$$

Per l'ellissoide con coordinate geografiche standard ($F = 0$):

$$\partial x/\partial \lambda \cdot \partial y/\partial \varphi - \partial x/\partial \varphi \cdot \partial y/\partial \lambda = N(\varphi)M(\varphi) \cos \varphi$$

Questa è un'equazione differenziale alle derivate parziali del primo ordine, più precisamente un'equazione di tipo Jacobiano. A differenza delle equazioni ellittiche per proiezioni conformi, questa singola equazione scalare permette grande libertà nella costruzione di proiezioni equivalenti.

Metodo delle caratteristiche: Possiamo parametrizzare le soluzioni introducendo coordinate ausiliarie (u, v) e scrivendo:

$$x = x(u, v) \quad y = y(u, v) \quad \lambda = \lambda(u, v) \quad \varphi = \varphi(u, v)$$

con:

$$\partial(x,y)/\partial(u,v) \cdot \partial(\lambda,\varphi)/\partial(u,v) = N(\varphi)M(\varphi) \cos \varphi$$

Scegliendo opportunamente la parametrizzazione ausiliaria, possiamo costruire infinite famiglie di proiezioni equivalenti.

5.2 Proiezioni Cilindriche Equivalenti

5.2.1 La Proiezione Cilindrica Equal-Area di Lambert

Questa è la più semplice proiezione equivalente, introdotta da Lambert nel 1772.

Derivazione: Cerchiamo una proiezione cilindrica della forma:

$$x = f(\lambda) \quad y = g(\varphi)$$

Per semplicità, scegliamo $x = R\lambda$ (meridiani equidistanti). La condizione di equivalenza diventa:

$$R \cdot g'(\varphi) = R^2 \cos \varphi$$

Quindi:

$$g'(\varphi) = R \cos \varphi$$

Integrando:

$$g(\varphi) = R \sin \varphi + C$$

Scegliendo $C = 0$ (l'equatore ha $y = 0$):

Cilindrica Equal-Area di Lambert (sfera):

$$x = R\lambda$$

$$y = R \sin \varphi$$

Proprietà:

1. Equivalenza esatta: $\sigma = hk = 1$ ovunque
2. Meridiani e paralleli sono linee rette perpendicolari
3. Fattori di scala:
 - $k = 1$ (lungo paralleli, scala esatta)
 - $h = 1/\cos \varphi$ (lungo meridiani)
4. Distorsione angolare massima:
 - All'equatore: $\omega = 0$ (nessuna distorsione)
 - A $\varphi = 60^\circ$: $\omega \approx 67^\circ$ (grave distorsione)
 - Ai poli: $\omega \rightarrow 90^\circ$ (distorsione infinita)

Estensione all'ellissoide: Sostituendo il seno della latitudine con una funzione authalica appropriata:

$$y = a \cdot q(\varphi)/q(\pi/2)$$

dove $q(\varphi)$ è la funzione authalica definita precedentemente.

5.2.2 Varianti Cilindriche: La Famiglia di Proiezioni di Gall-Peters

Possiamo generalizzare permettendo al cilindro di essere secante invece che tangente. Se il cilindro interseca la sfera ai paralleli $\pm\varphi_0$, la scala lungo questi paralleli è esatta.

Proiezione Cilindrica Equal-Area Generalizzata:

$$x = R\lambda \cos \varphi_0$$

$$y = R \sin \varphi / \cos \varphi_0$$

Casi speciali:

- $\varphi_0 = 0^\circ$: Lambert standard
- $\varphi_0 = 45^\circ$: Proiezione di Gall-Peters (1855/1973)
- $\varphi_0 = 30^\circ$: Proiezione di Behrmann (1910)
- $\varphi_0 = 37^\circ 24'$: Proiezione di Trystan Edwards

La scelta di φ_0 rappresenta un compromesso tra distorsione alle alte e basse latitudini.

5.3 Proiezioni Azimutali Equivalenti**5.3.1 Proiezione Azimutale Equal-Area di Lambert**

Questa proiezione, nota anche come Lambert Zenithal Equal-Area (LZEA), è la proiezione azimutale equivalente fondamentale.

Derivazione: Per una proiezione azimutale centrata al polo nord:

$$\rho = \rho(\varphi) \quad \theta = \lambda$$

In coordinate cartesiane:

$$x = \rho(\varphi) \cos \lambda \quad y = \rho(\varphi) \sin \lambda$$

I fattori di scala sono:

$$k = \rho/(R \cos \varphi) \text{ (tangenziale)} \quad h = |d\rho/d\varphi|/R \text{ (radiale)}$$

La condizione di equivalenza $hk = 1$ diventa:

$$(1/R)|d\rho/d\varphi| \cdot \rho/(R \cos \varphi) = 1$$

Semplificando:

$$\rho \, d\rho = R^2 \cos \varphi \, d\varphi$$

Integrando:

$$\rho^2/2 = R^2 \sin \varphi + C$$

Imponendo $\rho(\pi/2) = 0$ (il polo nord è al centro):

$$C = -R^2$$

Quindi:

$$\rho^2 = 2R^2(\sin \varphi - \sin(\pi/2)) = -2R^2(1 - \sin \varphi) = 2R^2(\sin \varphi - 1)$$

Ma per $\varphi < \pi/2$, $\sin \varphi < 1$, quindi consideriamo:

$$\rho^2 = 2R^2(1 - \sin \varphi)$$

Da cui:

$$\rho = \sqrt{2R^2(1 - \sin \varphi)} = R\sqrt{2(1 - \sin \varphi)}$$

Usando l'identità $1 - \sin \varphi = 2 \sin^2(\pi/4 - \varphi/2)$:

$$\rho = 2R \sin(\pi/4 - \varphi/2)$$

Proiezione Azimutale Equal-Area di Lambert (polare):

$$x = 2R \sin(\pi/4 - \varphi/2) \cos \lambda$$

$$y = 2R \sin(\pi/4 - \varphi/2) \sin \lambda$$

O equivalentemente:

$$\rho = 2R \sin(\pi/4 - \varphi/2) = 2R \cos(\pi/4 + \varphi/2) \theta = \lambda$$

Proprietà:

1. Equivalenza esatta ovunque
2. Simmetria radiale perfetta
3. Al centro (polo): $h = k = 1$ (nessuna distorsione)
4. All'equatore: $k = \sqrt{2}$, $h = 1/\sqrt{2}$
5. L'intera sfera (escludendo il polo opposto) è rappresentata in un cerchio di raggio $2R$

Confronto con la stereografica:

- Stereografica: conforme, $\rho = 2R \tan(\pi/4 - \varphi/2)$
- LZEA: equivalente, $\rho = 2R \sin(\pi/4 - \varphi/2)$
- Poiché $\sin x < \tan x$ per $0 < x < \pi/2$, LZEA comprime le regioni esterne rispetto alla stereografica

5.3.2 Proiezione Azimutale Obliqua ed Equatoriale

La proiezione può essere centrata in un punto arbitrario (λ_0, φ_0) usando rotazioni sferiche. Le formule generali coinvolgono:

1. Calcolo della distanza angolare c dal centro:

$$\cos c = \sin \varphi_0 \sin \varphi + \cos \varphi_0 \cos \varphi \cos(\lambda - \lambda_0)$$

2. Calcolo dell'azimut A :

$$\tan A = [\cos \varphi \sin(\lambda - \lambda_0)] / [\cos \varphi_0 \sin \varphi - \sin \varphi_0 \cos \varphi \cos(\lambda - \lambda_0)]$$

3. Le coordinate cartesiane:

$$x = 2R \sin(c/2) \sin A \quad y = 2R \sin(c/2) \cos A$$

Questo fornisce una proiezione equivalente centrata in qualsiasi punto desiderato.

5.4 Proiezioni Coniche Equivalenti

5.4.1 Proiezione Conica Equal-Area di Albers

La proiezione di Albers (1805) è la proiezione conica equivalente standard, molto usata per mappe di paesi che si estendono in direzione est-ovest.

Derivazione: Forma generale conica:

$$\rho = \rho(\varphi) \quad \theta = n\lambda$$

Fattori di scala:

$$k = \rho n / (R \cos \varphi) \quad h = |d\rho/d\varphi|/R$$

Condizione di equivalenza $hk = 1$:

$$(1/R)|d\rho/d\varphi| \cdot \rho n / (R \cos \varphi) = 1$$

Quindi:

$$\rho \, d\rho = -(R^2/n) \cos \varphi \, d\varphi$$

Il segno negativo appare perché ρ decresce quando φ aumenta (ci muoviamo verso il polo).
Integrando:

$$\rho^2/2 = -(R^2/n) \sin \varphi + C$$

Per determinare n e C , imponiamo che la scala sia esatta ai paralleli standard φ_1 e φ_2 :

$$k(\varphi_1) = 1: \rho(\varphi_1)n = R \cos \varphi_1 \quad k(\varphi_2) = 1: \rho(\varphi_2)n = R \cos \varphi_2$$

Sommando le due condizioni (dopo aver elevato al quadrato):

$$\rho^2(\varphi_1)n^2 + \rho^2(\varphi_2)n^2 = R^2(\cos^2\varphi_1 + \cos^2\varphi_2)$$

Dall'equazione integrale:

$$\rho^2(\varphi_1) = -(2R^2/n) \sin \varphi_1 + 2C \quad \rho^2(\varphi_2) = -(2R^2/n) \sin \varphi_2 + 2C$$

Sostituendo e risolvendo:

$$n = (\cos^2\varphi_1 + \cos^2\varphi_2) / (2(\sin \varphi_1 - \sin \varphi_2))$$

Oppure, più simmetricamente:

$$n = (\cos^2\varphi_1 + \cos^2\varphi_2) / (\sin \varphi_1 - \sin \varphi_2) \cdot (1/2)$$

Una volta determinato n :

$$C = [\rho^2(\varphi_1) + (2R^2/n) \sin \varphi_1] / 2$$

La formula finale per ρ è:

$$\rho = (1/n) \sqrt{[C - 2(R^2/n) \sin \varphi]}$$

Formule complete per Albers (sfera):

$$n = (\sin \varphi_1 + \sin \varphi_2) / 2$$

$$C = \cos^2 \varphi_1 + 2n \sin \varphi_1$$

$$\rho = (R/n) \sqrt{C - 2n \sin \varphi}$$

$$\theta = n\lambda$$

$$x = \rho \sin \theta$$

$$y = \rho_0 - \rho \cos \theta$$

dove $\rho_0 = (R/n) \sqrt{C - 2n \sin \varphi_0}$ per una latitudine di origine φ_0 .

Proprietà:

1. Equivalenza esatta
2. Scala esatta ai due paralleli standard
3. Meridiani sono raggi equidistanti
4. Paralleli sono archi di cerchi concentrici non equidistanti
5. Distorsione minima per regioni est-ovest a latitudini medie

Estensione all'ellissoide: Le formule diventano più complesse, coinvolgendo la funzione authalica $q(\varphi)$:

$$n = (N_1 \cos \varphi_1 - N_2 \cos \varphi_2) / (q(\varphi_1) - q(\varphi_2))$$

dove $N_1 = N(\varphi_1)$ e $N_2 = N(\varphi_2)$ sono i raggi di curvatura.

5.5 Proiezioni Pseudocilindriche Equivalenti

Le proiezioni pseudocilindriche mantengono i paralleli come linee rette parallele, ma i meridiani diventano curve (non più rette parallele).

5.5.1 Proiezione Sinusoidale (Sanson-Flamsteed)

Una delle più antiche proiezioni equivalenti, risalente al XVI secolo.

Derivazione: Vogliamo:

- Paralleli come linee rette: $y = g(\varphi)$
- Meridiani come curve: $x = f(\lambda, \varphi)$
- Meridiano centrale ($\lambda = 0$) retto: $f(0, \varphi) = 0$
- Scala esatta lungo paralleli: $\partial x / \partial \lambda = R \cos \varphi$

Integrando la quarta condizione:

$$x = R\lambda \cos \varphi$$

Per l'equivalenza:

$$\partial x / \partial \lambda \cdot \partial y / \partial \varphi = R^2 \cos \varphi$$

Quindi:

$$R \cos \varphi \cdot g'(\varphi) = R^2 \cos \varphi g'(\varphi) = R g(\varphi) = R\varphi$$

Proiezione Sinusoidale:

$$x = R\lambda \cos \varphi$$

$$y = R\varphi$$

Proprietà:

1. Equivalente
2. Meridiano centrale e equatore non distorti
3. Scala esatta lungo tutti i paralleli
4. I meridiani sono sinusoidi: per λ fissato, $x = R\lambda \cos(y/R)$
5. Distorsione cresce rapidamente lontano dal meridiano centrale

Versione interrotta: Per ridurre la distorsione, Goode (1916) propose di "interrompere" la proiezione, usando diversi meridiani centrali per diversi lobi. La "proiezione omolografica di Goode" usa tipicamente 6 lobi per i continenti, minimizzando la distorsione nelle masse terrestri a scapito degli oceani.

5.5.2 Proiezione di Mollweide (Omolografica)

La proiezione di Mollweide (1805) rappresenta la sfera in un'ellisse, producendo un aspetto più gradevole della sinusoidale.

Derivazione: Vogliamo rappresentare la sfera in un'ellisse con semiassi $a = 2R\sqrt{2}$ (orizzontale) e $b = R\sqrt{2}$ (verticale).

Forma generale:

$$x = (2R\sqrt{2}/\pi)\lambda \cos \theta \quad y = R\sqrt{2} \sin \theta$$

dove $\theta = \theta(\varphi)$ è una funzione ausiliaria da determinare.

Condizione di equivalenza: L'area tra l'equatore e latitudine φ sulla sfera è:

$$A_{\text{sfera}} = 2\pi R^2 \sin \varphi$$

L'area corrispondente nell'ellisse è:

$$A_{\text{ellisse}} = \int_0^\theta (2R\sqrt{2}/\pi) \cdot 2\pi \cdot R\sqrt{2} \cos \theta' d\theta' = 4R^2\theta + 4R^2 \sin \theta \cos \theta = 4R^2(\theta + (1/2) \sin 2\theta)$$

Uguagliando:

$$2\pi R^2 \sin \varphi = 4R^2(\theta + (1/2) \sin 2\theta)$$

Semplificando:

$$2\theta + \sin 2\theta = \pi \sin \varphi$$

Questa è un'equazione trascendente per θ in funzione di φ . Non ha soluzione in forma chiusa e richiede metodi numerici (tipicamente Newton-Raphson).

Proiezione di Mollweide:

$$2\theta + \sin 2\theta = \pi \sin \varphi \quad (\text{risolvere per } \theta)$$

$$x = (2\sqrt{2} R/\pi)\lambda \cos \theta$$

$$y = \sqrt{2} R \sin \theta$$

Proprietà:

1. Equivalente
2. Forma ellittica gradevole
3. Meridiano centrale e equatore rettilinei non distorti
4. I poli sono rappresentati come punti (vertici dell'ellisse)
5. Bilanciamento migliore della distorsione rispetto alla sinusoidale

5.5.3 Proiezione di Eckert IV

Una delle sei proiezioni proposte da Max Eckert (1906), tutte pseudocilindriche. Eckert IV è particolarmente popolare.

Formule:

$$\theta + \sin \theta \cos \theta + 2 \sin \theta = (2 + \pi/2) \sin \varphi \quad (\text{risolvere per } \theta)$$

$$x = (2R\lambda/\sqrt{\pi(4+\pi)})(1 + \cos \theta)$$

$$y = 2R\sqrt{\pi/(4+\pi)} \sin \theta$$

La forma è un'ellisse con rapporto assiale 2:1 (più allungata di Mollweide).

5.6 Teorema di Chebyshev sull'Equivalenza Ottimale

Per regioni di forma data, esiste una proiezione equivalente che minimizza la distorsione angolare massima?

Teorema 5.1 (Chebyshev, 1856): Per una regione semplicemente connessa sulla sfera, esiste un'unica proiezione equivalente che minimizza:

$$\max_{\{P \in D\}} \omega_{\max}(P)$$

dove $\omega_{\max}(P)$ è la massima distorsione angolare nel punto P.

Questa proiezione ottimale è caratterizzata dalla condizione che la massima distorsione angolare sia costante sul bordo della regione.

La dimostrazione coinvolge teoria del potenziale e calcolo delle variazioni. In pratica, queste proiezioni ottimali sono raramente usate perché:

1. Dipendono fortemente dalla forma specifica della regione
2. Sono computazionalmente complesse
3. Non hanno proprietà geometriche semplici (meridiani/paralleli non hanno forme standard)

Tuttavia, il teorema è importante teoricamente perché stabilisce limiti inferiori sulla distorsione inevitabile.

Capitolo 6: Proiezioni Equidistanti e Proiezioni di Compromesso

6.1 Teoria delle Proiezioni Equidistanti

Le proiezioni equidistanti preservano le distanze lungo determinate direzioni o da determinati punti. A differenza delle proiezioni conformi ed equivalenti, non esiste una teoria unificata per tutte le proiezioni equidistanti, poiché la proprietà di equidistanza può essere specificata in molti modi diversi.

6.1.1 Impossibilità dell'Equidistanza Globale

Teorema 6.1: Non esiste una proiezione che preservi tutte le distanze tra tutti i punti.

Dimostrazione: Supponiamo per assurdo che esista una proiezione $\phi: S^2 \rightarrow R^2$ che preserva tutte le distanze geodetiche. Consideriamo tre punti A, B, C sulla sfera che formano un triangolo sferico con lati di lunghezza a, b, c e angoli α, β, γ .

Per la geometria sferica, vale: $\alpha + \beta + \gamma = \pi + E$

dove $E > 0$ è l'eccesso sferico, dato da $E = \text{Area}/(R^2)$.

Se ϕ preserva tutte le distanze, l'immagine di ABC nel piano è un triangolo euclideo con gli stessi lati a, b, c. Ma in geometria euclidea: $\alpha' + \beta' + \gamma' = \pi$

Questo implica $E = 0$, quindi Area = 0, il che è possibile solo per triangoli degeneri. Per triangoli non degeneri, abbiamo una contraddizione. $\square$

Corollario 6.2: Ogni proiezione equidistante può preservare distanze solo lungo determinate famiglie di curve, non ovunque.

6.1.2 Proiezione Azimutale Equidistante

La proiezione azimutale equidistante preserva le distanze dal centro della proiezione e le direzioni (azimut) dal centro.

Derivazione: Per una proiezione azimutale centrata al polo nord:

$$\rho = \rho(\varphi) \quad \theta = \lambda$$

La distanza geodetica dal polo nord a un punto di latitudine φ è:

$$d = R(\pi/2 - \varphi)$$

Per equidistanza radiale, vogliamo $\rho = d$:

$$\rho = R(\pi/2 - \varphi)$$

Proiezione Azimutale Equidistante (polare):

$$\rho = R(\pi/2 - \varphi)$$

$$\theta = \lambda$$

$$x = R(\pi/2 - \varphi) \cos \lambda$$

$$y = R(\pi/2 - \varphi) \sin \lambda$$

Proprietà:

1. Distanze dal centro preservate esattamente
2. Direzioni dal centro (azimut) preservate
3. Al centro: $h = k = 1$ (nessuna distorsione)
4. Lungo raggi: $h = 1$ (scala esatta)
5. Tangenzialmente: $k = (\pi/2 - \varphi)/\sin(\pi/2 - \varphi) = (\pi/2 - \varphi)/\cos \varphi$
6. L'intera sfera è rappresentata in un cerchio di raggio πR

Fattori di scala:

- Radiale: $h = 1$
- Tangenziale: $k = \rho/(R \cos \varphi) = (\pi/2 - \varphi)/\cos \varphi$

All'equatore ($\varphi = 0$): $k = \pi/2 \approx 1.571$ (57% di allungamento) Al polo opposto ($\varphi = -\pi/2$): $k \rightarrow \infty$

Versione obliqua: Centrata in un punto arbitrario (λ_0, φ_0), la distanza angolare c è:

$$\cos c = \sin \varphi_0 \sin \varphi + \cos \varphi_0 \cos \varphi \cos(\lambda - \lambda_0)$$

E l'azimut A :

$$\sin A = \cos \varphi \sin(\lambda - \lambda_0)/\sin c \quad \cos A = [\cos \varphi_0 \sin \varphi - \sin \varphi_0 \cos \varphi \cos(\lambda - \lambda_0)]/\sin c$$

Le coordinate sono:

$$x = Rc \cos A$$

$$y = Rc \sin A$$

Applicazioni:

1. Mappe radio e radar (distanze e direzioni da una stazione)
2. Rotte di volo dalla capitale di un paese
3. Mappe sismiche (distanze da epicentro)
4. Logo delle Nazioni Unite (centrato al polo nord)

6.1.3 Proiezione Cilindrica Equidistante (Plate Carrée)

La più semplice di tutte le proiezioni, nota anche come proiezione equirettangolare.

Formule:

$$x = R\lambda$$

$$y = R\varphi$$

È semplicemente una mappatura diretta delle coordinate sferiche (λ , φ) sul piano cartesiano.

Proprietà:

1. Meridiani e paralleli sono griglie rettangolari uniformi
2. Distanze lungo meridiani: $h = 1$ (esatte)
3. Distanze lungo paralleli: $k = 1/\cos \varphi$ (distorte eccetto all'equatore)
4. Non conforme, non equivalente, non equidistante globalmente
5. Estremamente semplice computazionalmente

Variante: Plate Carrée con parallelo standard φ_0 :

$$x = R\lambda \cos \varphi_0$$

$$y = R\varphi$$

La scala è esatta lungo il parallelo φ_0 (e $-\varphi_0$).

6.1.4 Proiezione Conica Equidistante

Preserva distanze lungo i meridiani.

Formule:

$$n = (\cos \varphi_1 - \cos \varphi_2) / (\varphi_2 - \varphi_1)$$

$$G = \cos \varphi_1 / n + \varphi_1$$

$$\rho = R(G - \varphi)$$

$$\theta = n\lambda$$

$$x = \rho \sin \theta$$

$$y = \rho_0 - \rho \cos \theta$$

dove φ_1 e φ_2 sono i paralleli standard.

Proprietà:

1. $h = 1$ lungo tutti i meridiani
2. $k = 1$ ai paralleli standard
3. Paralleli sono archi di cerchio equidistanti
4. Usata per atlanti e mappe di paesi a sviluppo est-ovest

6.2 Proiezioni di Compromesso

Le proiezioni di compromesso non preservano esattamente angoli, aree o distanze, ma cercano di minimizzare la distorsione complessiva secondo qualche criterio.

6.2.1 Proiezione di Robinson

Sviluppata nel 1963 da Arthur Robinson per la National Geographic Society, questa proiezione divenne lo standard per le mappe del mondo fino al 1998.

Caratteristiche:

- Pseudocilindrica (paralleli rettilinei, meridiani curvi)
- Non ha espressione analitica chiusa
- Definita da tabelle interpolate
- Progettata per "bell'aspetto" più che per rigore matematico

Tabella di Robinson (valori tabulati per y e lunghezza del parallelo):

Latitudine	y/R	Lunghezza parallelo
0°	0.0000	1.0000
5°	0.0620	0.9986
10°	0.1240	0.9954
...	...	...
85°	0.8960	0.0620
90°	0.9360	0.0000

Le coordinate sono calcolate per interpolazione (tipicamente spline cubiche).

Proprietà:

1. Bilanciamento visivo tra diverse forme di distorsione
2. Paralleli distanziati in modo decrescente verso i poli
3. Distorsione massima contenuta (ma non ottimizzata matematicamente)
4. Poli rappresentati come linee (lunghezza $\approx 0.53 \times$ equatore)

6.2.2 Proiezione di Winkel Tripel

Adottata dalla National Geographic Society nel 1998 in sostituzione della Robinson.

Derivazione: Media aritmetica tra la proiezione azimutale equidistante (centrata all'equatore) e la Plate Carrée con parallelo standard φ_1 :

$$x = (x_{AE} + x_{PC})/2 \quad y = (y_{AE} + y_{PC})/2$$

Più precisamente, con $\varphi_1 = \arccos(2/\pi) \approx 50.467^\circ$:

$$\alpha = \arccos(\cos \varphi \cos \lambda/2)$$

$$x = (1/2)[R\lambda \cos \varphi_1 + 2R \cos \varphi \sin(\lambda/2)/\text{sinc}(\alpha)]$$

$$y = (1/2)[R\varphi + R \sin \varphi/\text{sinc}(\alpha)]$$

dove $\text{sinc}(\alpha) = \sin(\alpha)/\alpha$.

Proprietà:

1. Riduzione della distorsione rispetto a Robinson
2. Forma più arrotondata agli angoli
3. Migliore compromesso secondo metriche quantitative (RMS della distorsione)
4. Espressione analitica esplicita

6.2.3 Proiezione di Miller Cylindrical

Una modifica della proiezione di Mercatore che evita la distorsione estrema ai poli.

Formule:

$$x = R\lambda$$

$$y = (5R/4) \ln \tan(\pi/4 + 2\varphi/5)$$

Essenzialmente, si applica Mercatore alla latitudine 0.8φ invece di φ , poi si scala il risultato.

Proprietà:

1. Simile a Mercatore ma con poli rappresentabili (fino a $\pm\pi/2$)
2. Non conforme (gli angoli sono leggermente distorti)
3. Non equivalente
4. Compromesso tra l'aspetto familiare di Mercatore e distorsione ridotta
5. Usata per mappe murali e presentazioni

6.2.4 Criteri Quantitativi per Proiezioni di Compromesso

Diversi indici sono stati proposti per quantificare la "bontà" di una proiezione:

Indice di Airy-Kavrayskiy: $I_{AK} = \iint_D \left[(h-1)^2 + (k-1)^2 \right] dA$

Misura lo scostamento dai fattori di scala unitari.

Indice di Airy modificato: $I_A = \iint_D (h^2 + k^2 - 2) dA$

Equivalente per piccole distorsioni.

Indice di Peters: $I_P = \iint_D |hk - 1| dA$

Misura lo scostamento dall'equivalenza.

Indice combinato di Goldberg-Gott: $I_{GG} = \sqrt{\alpha I_a^2 + \beta I_d^2 + \gamma I_s^2}$

dove I_a misura distorsione areale, I_d distorsione di distanze, I_s distorsione di forma, e α , β , γ sono pesi.

6.3 Proiezioni Policoniche

Le proiezioni policoniche usano una famiglia di coni, ciascuno tangente a un parallelo diverso.

6.3.1 Proiezione Policonica Ordinaria

Ogni parallelo è l'arco di un cerchio che sarebbe ottenuto sviluppando un cono tangente a quel parallelo.

Formule (per la sfera):

Per ogni latitudine φ :

- Raggio del parallelo: $\rho(\varphi) = R \cot \varphi$
- Angolo al vertice del cono: $\theta = \lambda \sin \varphi$
- Distanza dal vertice lungo il meridiano da φ_0 a φ :

$$E = R \int_{\varphi_0}^{\varphi} \sec \varphi' d\varphi' = R \ln |\sec \varphi + \tan \varphi| + \text{costante}$$

Le coordinate finali:

$$x = \rho(\varphi) \sin \theta = (R \cot \varphi) \sin(\lambda \sin \varphi)$$

$$y = E + \rho(\varphi)[1 - \cos \theta] = R[\ln \tan(\pi/4 + \varphi/2)] + R \cot \varphi[1 - \cos(\lambda \sin \varphi)]$$

Proprietà:

1. Meridiano centrale è una linea retta non distorta
2. Tutti i paralleli hanno scala esatta

3. Non conforme, non equivalente
4. Distorsione cresce rapidamente lontano dal meridiano centrale
5. Usata dallo U.S. Coast and Geodetic Survey nel XIX secolo

6.3.2 Proiezione Policonica Modificata (IMW)

Sviluppata per l'International Map of the World (IMW) al milionesimo. Usa una formula modificata per ridurre la distorsione nei fogli rettangolari standard.

6.4 Proiezioni Retroazimutali

Una classe insolita di proiezioni dove ogni punto sulla mappa indica la direzione corretta verso un punto centrale di riferimento.

6.4.1 Proiezione di Craig Retroazimutale

Tutte le linee rette passanti per il punto centrale sulla mappa puntano nella direzione corretta per tornare a quel punto sulla sfera.

Formule (centrata al polo nord):

$$\rho = 2R \tan(c/2)$$

$$\theta = \lambda - \pi$$

dove c è la distanza angolare dal centro.

Proprietà:

1. Direzione inversa preservata ovunque
2. Conforme solo al centro
3. Usata per navigazione "verso casa" da qualsiasi punto

Capitolo 7: Datum Geodetici e Sistemi di Riferimento

7.1 Concetto di Datum Geodetico

Un datum geodetico è un sistema di riferimento che definisce:

1. Un ellissoide di riferimento specifico
2. L'orientazione e posizione di questo ellissoide rispetto alla Terra fisica
3. Un punto di origine e direzioni degli assi

Definizione 7.1: Un datum geodetico è specificato da:

- Parametri dell'ellissoide (a , f o a , b)
- Coordinate di un punto fondamentale
- Azimut di una direzione di riferimento

- Altezza del geoide in quel punto

7.2 Ellissoidi di Riferimento Storici e Moderni

7.2.1 Ellissoidi Storici

Bessel 1841:

- $a = 6,377,397.155 \text{ m}$
- $f^{-1} = 299.1528128$
- Uso: Germania, Giappone, Indonesia
- Basato su misurazioni dell'arco prussiano-russo

Clarke 1866:

- $a = 6,378,206.4 \text{ m}$
- $f^{-1} = 294.978698$
- Uso: Nord America (NAD27)
- Basato su triangolazioni del North American Datum

International 1924 (Hayford):

- $a = 6,378,388.0 \text{ m}$
- $f^{-1} = 297.0$
- Uso: Europa, Sud America
- Adottato dall'International Union of Geodesy and Geophysics

GRS 1967 (Geodetic Reference System):

- $a = 6,378,160.0 \text{ m}$
- $f^{-1} = 298.247167$
- Transizione verso sistemi geocentrici

7.2.2 Ellissoidi Moderni

GRS 1980 (Geodetic Reference System 1980):

- $a = 6,378,137.0 \text{ m}$
- $f^{-1} = 298.257222101$
- $GM = 3.986005 \times 10^{14} \text{ m}^3/\text{s}^2$ (massa gravitazionale della Terra)
- $\omega = 7.292115 \times 10^{-5} \text{ rad/s}$ (velocità angolare)
- Basato su dati satellitari e teoria del potenziale gravitazionale

WGS 84 (World Geodetic System 1984):

- $a = 6,378,137.0 \text{ m}$
- $f^{-1} = 298.257223563$
- $GM = 3.986004418 \times 10^{14} \text{ m}^3/\text{s}^2$

- $\omega = 7.292115 \times 10^{-5}$ rad/s
- Sistema GPS globale
- Praticamente identico a GRS80 (differenza in f solo alla ottava cifra decimale)

Le differenze tra GRS80 e WGS84 sono:

- $\Delta a = 0$ m
- $\Delta f \approx 10^{-11}$ (completamente trascurabile nella pratica)
- Differenze nelle coordinate < 1 mm globalmente

7.3 Trasformazioni tra Datum

Quando si lavora con dati provenienti da diversi datum, è necessario applicare trasformazioni di coordinate.

7.3.1 Trasformazione a Tre Parametri

La trasformazione più semplice coinvolge solo una traslazione:

$$X_2 = X_1 + \Delta X$$

$$Y_2 = Y_1 + \Delta Y$$

$$Z_2 = Z_1 + \Delta Z$$

dove (X, Y, Z) sono coordinate cartesiane geocentriche e (ΔX , ΔY , ΔZ) sono le componenti di traslazione.

Esempio: NAD27 → NAD83 negli USA continentali:

- $\Delta X \approx -9$ m
- $\Delta Y \approx +160$ m
- $\Delta Z \approx +176$ m

(valori approssimati; variano localmente)

7.3.2 Trasformazione di Helmert a Sette Parametri

La trasformazione più completa include:

- 3 traslazioni: (ΔX , ΔY , ΔZ)
- 3 rotazioni: (R_x , R_y , R_z)
- 1 fattore di scala: s

Formula matriciale:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} + (1+s)R \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}$$

dove R è la matrice di rotazione:

$$R = \begin{bmatrix} 1 & R_y & -R_x \\ -R_y & 1 & R_x \\ R_y & -R_x & 1 \end{bmatrix}$$

Per piccole rotazioni (in radianti), questa approssimazione lineare è valida.

Esempio: ETRS89 $\rightarrow$ WGS84:

- Traslazioni: ordine di cm
- Rotazioni: ordine di milliarcsec (10^{-9} rad)
- Scala: ordine di 10^{-9}

7.3.3 Trasformazione di Molodensky

Per trasformazioni dirette di coordinate geodetiche (λ, φ, h) senza passare per coordinate cartesiane:

$$\Delta\varphi = [-\Delta X \sin \varphi \cos \lambda - \Delta Y \sin \varphi \sin \lambda + \Delta Z \cos \varphi + \Delta a(N/a)e^2 \sin \varphi \cos \varphi + \Delta f(M/a + N/a)\sin \varphi \cos \varphi]/(M + h)$$

$$\Delta\lambda = [-\Delta X \sin \lambda + \Delta Y \cos \lambda]/[(N + h) \cos \varphi]$$

$$\Delta h = \Delta X \cos \varphi \cos \lambda + \Delta Y \cos \varphi \sin \lambda + \Delta Z \sin \varphi - \Delta a(a/N) + \Delta f(b/N)(N/a)\sin^2 \varphi$$

dove:

- $\Delta a = a_2 - a_1$ (differenza nei semiassi maggiori)
- $\Delta f = f_2 - f_1$ (differenza negli schiacciamenti)
- M, N sono i raggi di curvatura

Queste formule evitano la conversione intermedia in coordinate cartesiane, risultando più efficienti computazionalmente.

7.4 Il Geoide e la Separazione Geoide-Ellissoide

7.4.1 Definizione del Geoide

Il geoide è la superficie equipotenziale del campo gravitazionale terrestre che meglio approssima il livello medio del mare.

Definizione matematica: Il geoide è definito da:

$$W(X, Y, Z) = W_0$$

dove W è il potenziale gravitazionale efficace (inclusi effetti rotazionali):

$$W = V + \Phi_{\text{centrifugo}}$$

con:

- V : potenziale gravitazionale newtoniano
- $\Phi_{\text{centrifugo}} = -(1/2)\omega^2(X^2 + Y^2)$

Il valore W_0 è scelto per far coincidere il geoide con il livello medio del mare.

7.4.2 Ondulazione del Geoide

L'ondulazione del geoide N è la distanza verticale (lungo la normale all'ellissoide) tra geoide ed ellissoide:

$$N = h - H$$

dove:

- h : altezza ellissoidale (dal GPS)
- H : altezza ortometrica (sul livello del mare, dal livellamento)
- N : ondulazione del geoide

Variazione globale di N :

- Minimo: circa -106 m (Oceano Indiano, al largo dello Sri Lanka)
- Massimo: circa +85 m (Nuova Guinea)
- Range totale: circa 190 m

Localmente, N varia tipicamente di 10-50 m su distanze di centinaia di km.

7.4.3 Modelli Globali del Geoide

I modelli moderni del geoide sono espressi come sviluppi in armoniche sferiche:

$$N(\theta, \lambda) = \sum_{n=2}^{N_{\text{ax}}} \sum_{m=0}^n P_{n,m}(\sin \varphi) \cdot [C_{n,m} \cdot \cos(m\lambda) + S_{n,m} \cdot \sin(m\lambda)]$$

dove:

- $P_{n,m}$: funzioni di Legendre associate
- $C_{n,m}, S_{n,m}$: coefficienti armonici (determinati da misure satellitari e terrestri)
- N_{ax} : grado massimo dello sviluppo

Modelli principali:

1. **EGM96** (Earth Gravitational Model 1996):
 - Grado 360×360 (risoluzione ~55 km)
 - Accuratezza: ± 0.5 -1.0 m globalmente

2. **EGM2008:**

- Grado 2190×2190 (risoluzione ~ 9 km)
- Accuratezza: $\pm 0.1-0.2$ m in aree ben misurate
- ~ 5 milioni di coefficienti

3. **EIGEN-6C4:**

- Grado 2190×2190
- Incorpora dati da GRACE, GOCE, LAGEOS

4. **XGM2019e:**

- Modello più recente (2019)
- Combinazione di dati satellitari e terrestri
- Massima risoluzione e accuratezza attuale

7.4.4 Calcolo Pratico dell'Ondulazione

Per applicazioni pratiche, i modelli armonici sferici fino a grado 2190 sono computazionalmente intensivi. Metodi più efficienti includono:

1. **Interpolazione da griglie:** I valori di N sono precalcolati su una griglia regolare e interpolati
2. **Approssimazioni locali:** Polinomi di basso grado su regioni limitate
3. **Software specializzato:** Librerie come GeographicLib, PROJ forniscono accesso rapido

Accuratezza richiesta:

- Navigazione generale: $\pm 1-2$ m
- Rilievi topografici: $\pm 0.1-0.5$ m
- Geodesia di precisione: $\pm 0.01-0.05$ m
- Riferimenti verticali nazionali: $\pm 0.001-0.01$ m

7.5 Sistemi di Riferimento Terrestri Moderni

7.5.1 ITRS (International Terrestrial Reference System)

L'ITRS è il sistema di riferimento globale standard definito dall'IERS (International Earth Rotation and Reference Systems Service).

Caratteristiche:

- Origine: centro di massa della Terra (inclusi oceani e atmosfera)
- Asse Z: direzione del polo convenzionale terrestre (CTP)
- Asse X: intersezione del piano meridiano di riferimento dell'IERS e piano equatoriale
- Asse Y: completa il sistema destrorso

Realizzazioni pratiche: ITRF (International Terrestrial Reference Frame)

- ITRF89, ITRF90, ITRF91, ..., ITRF2020
- Ogni realizzazione fornisce coordinate precise per una rete di stazioni di riferimento
- Accuratezza: sub-centimetrica per posizioni, mm/anno per velocità

ITRF2020 (più recente):

- ~1600 stazioni di riferimento
- Combina dati da: VLBI, SLR, GPS, DORIS, GNSS
- Epoca di riferimento: 2015.0
- Include velocità delle stazioni (movimento delle placche tettoniche)

7.5.2 WGS84 e sue Realizzazioni

Il World Geodetic System 1984 è il sistema usato dal GPS.

Evoluzione di WGS84:

- WGS84 (originale, 1987): accuratezza ~1-2 m
- WGS84 (G730, 1994): accuratezza ~10 cm
- WGS84 (G873, 1997): allineato con ITRF94
- WGS84 (G1150, 2002): allineato con ITRF2000
- WGS84 (G1674, 2012): allineato con ITRF2008
- WGS84 (G1762, 2013): allineato con ITRF2008
- WGS84 (G2139, 2021): allineato con ITRF2014

Ogni realizzazione migliora l'accuratezza e l'allineamento con ITRF.

Compatibilità pratica: Per la maggior parte delle applicazioni (precisione dm-m), WGS84 e ITRF correnti sono intercambiabili. Differenze significative (cm) emergono solo in geodesia di alta precisione.

7.5.3 Sistemi Regionali

ETRS89 (European Terrestrial Reference System 1989):

- Coincidente con ITRS all'epoca 1989.0
- Fissato alla placca euroasiatica (non aggiornato per la deriva continentale)
- Usato come riferimento geodetico standard in Europa

NAD83 (North American Datum 1983):

- Basato su GRS80
- Originariamente coincidente con WGS84, ma divergenze accumulate nel tempo
- NAD83(CORS) è la realizzazione moderna, continuamente aggiornata
- Differenze con WGS84: fino a 1-2 m in alcune regioni

Altri sistemi regionali:

- SIRGAS (Sistema de Referencia Geocéntrico para las Américas) - Sud America

- GDA2020 (Geocentric Datum of Australia 2020)
- JGD2011 (Japanese Geodetic Datum 2011)
- NZGD2000 (New Zealand Geodetic Datum 2000)

7.6 Coordinate UTM e Sistemi Planimetrici

7.6.1 Sistema UTM Completo

Il sistema Universal Transverse Mercator divide il mondo in 60 zone di 6° di longitudine.

Struttura delle zone UTM:

- Zone numerate da 1 a 60, procedendo verso est
- Zona 1: 180°W - 174°W
- Ogni zona ha meridiano centrale a $\lambda_0 = -183^\circ + 6n^\circ$, dove n è il numero della zona
- Copertura latitudinale: 80°S - 84°N

Bande latitudinali (solo per notazione, non per calcoli):

- Designate con lettere da C (80°S-72°S) a X (72°N-84°N)
- Lettere I e O omesse per evitare confusione con 1 e 0
- Esempio: "33T" indica zona 33, banda T (40°N-48°N)

Parametri di proiezione:

- Proiezione: Mercatore Trasversa
- Ellissoide: WGS84 (o GRS80, praticamente identico)
- Fattore di scala al meridiano centrale: $k_0 = 0.9996$
- False Easting: 500,000 m
- False Northing: 0 m (emisfero nord), 10,000,000 m (emisfero sud)

Formule complete per UTM (serie di Kruger fino al 6° ordine in λ):

Definiamo le quantità ausiliarie:

$$n = f/(2-f) \quad (\text{terzo schiacciamento})$$

$$A = a/(1+n) \cdot [1 + n^2/4 + n^4/64 + \dots] \quad (\text{raggio della sfera ausiliaria})$$

Latitudine conforme:

$$\xi = \arctan(\sinh(\eta))$$

$$\eta = \operatorname{arctanh}(\sin \varphi) - e \cdot \operatorname{arctanh}(e \sin \varphi)$$

Sviluppo in serie per le coordinate:

$$E = E_0 + k_0 A \cdot (\eta + \sum \alpha_j \sin(2j\xi) \sinh(2j\eta))$$

$$N = N_0 + k_0 A \cdot (\xi + \sum \alpha_j \cos(2j\xi) \cosh(2j\eta))$$

dove i coefficienti α_i sono funzioni di n :

$$\alpha_1 = n/2 - 2n^2/3 + 5n^3/16 + 41n^4/180 - \dots$$

$$\alpha_2 = 13n^2/48 - 3n^3/5 + 557n^4/1440 + \dots$$

$$\alpha_3 = 61n^3/240 - 103n^4/140 + \dots$$

$$\alpha_4 = 49561n^4/161280 - \dots$$

Precisione: Le serie fino all'ordine 6 forniscono accuratezza millimetrica all'interno della zona UTM.

7.6.2 Convergenza dei Meridiani in UTM

In una proiezione Mercatore Trasversa, i meridiani non sono paralleli ma convergono verso il meridiano centrale. La convergenza γ (angolo tra il nord griglia e il nord geografico) è:

$$\gamma = \arctan(\sinh(\eta)) \approx (\lambda - \lambda_0) \sin \varphi + (\lambda - \lambda_0)^3 \cos^2 \varphi \sin \varphi / 3 + \dots$$

Per piccole differenze di longitudine:

$$\gamma \approx (\lambda - \lambda_0) \sin \varphi$$

Esempio: Punto a 40°N, 3° est del meridiano centrale:

$$\gamma \approx 3^\circ \times \sin(40^\circ) \approx 3^\circ \times 0.643 \approx 1.93^\circ \approx 1^\circ 56'$$

Questa convergenza deve essere considerata quando si orienta una mappa o si misura azimut.

7.6.3 Fattore di Scala in UTM

Il fattore di scala k varia con la distanza dal meridiano centrale:

$$k = k_0 [1 + (E - E_0)^2 / (2R^2) + \dots]$$

Dove R è il raggio medio di curvatura.

Valori tipici:

- Al meridiano centrale: $k = 0.9996$ (0.04% di riduzione)
- A 180 km dal meridiano centrale: $k \approx 1.0000$
- Ai bordi della zona ($\pm 3^\circ$): $k \approx 1.0004$ (0.04% di allungamento)

Distorsione lineare massima: circa 1:2500 all'interno di una zona UTM

7.7 Sistemi di Coordinate Nazionali

Molti paesi hanno sviluppato sistemi di coordinate planimetriche ottimizzati per il loro territorio.

7.7.1 Sistema Gauss-Boaga (Italia)

Il sistema storico italiano, basato su Mercatore Trasversa.

Caratteristiche:

- Ellissoide: Internazionale 1924 (Hayford)
- Due fusi:
 - Fuso Ovest: meridiano centrale 9°E (Roma)
 - Fuso Est: meridiano centrale 15°E
- Ampiezza dei fusi: 6° ($\pm 3^\circ$ dal meridiano centrale)
- $k_0 = 0.9996$
- False Easting: 1,500,000 m (Ovest), 2,520,000 m (Est)
- False Northing: 0 m

Coordinate Gauss-Boaga: Un punto in Italia ha due rappresentazioni:

- Coordinate geografiche (φ , λ) su International 1924
- Coordinate planimetriche (E, N) nel fuso appropriato

Precisione: ± 1 -2 m per scopi topografici tradizionali

7.7.2 Sistema UTM-WGS84/ETRF2000 (Italia moderna)

Dal 2011, l'Italia ha adottato un sistema basato su ETRF2000.

Transizione:

- Datum: ETRF2000 (epoca 2008.0)
- Proiezione: UTM (zone 32, 33, 34)
- Ellissoide: GRS80

Trasformazione da Gauss-Boaga a UTM-ETRF2000: Richiede:

1. Conversione coordinate Gauss-Boaga $\rightarrow$ geografiche su Int. 1924
2. Trasformazione datum Int. 1924 $\rightarrow$ ETRF2000 (Helmert a 7 parametri)
3. Proiezione geografiche ETRF2000 $\rightarrow$ UTM

Parametri di trasformazione (medi per l'Italia):

$$\Delta X \approx -225 \text{ m}$$

$$\Delta Y \approx +65 \text{ m}$$

$\Delta Z \approx +9 \text{ m}$
 $R_x \approx -0.0005''$
 $R_y \approx -0.0047''$
 $R_y \approx +0.0063''$
 $s \approx +1.0 \text{ ppm}$

Gli spostamenti planimetrici risultanti sono tipicamente 150-250 m.

7.7.3 State Plane Coordinate System (USA)

Sistema complesso che divide ogni stato USA in zone ottimizzate.

Caratteristiche:

- Proiezioni usate: Mercatore Trasversa (per stati stretti nord-sud), Conica Conforme di Lambert (per stati larghi est-ovest), Obliqua di Mercatore (solo Alaska)
- Datum: NAD83 (moderno), NAD27 (storico)
- Unità: metri (modernamente) o piedi USA (storicamente)
- Oltre 120 zone per coprire tutti gli stati

Esempio: California:

- 6 zone, tutte coniche conformi di Lambert
- Zone 1-6 coprono lo stato da nord a sud
- Zona 5 (Los Angeles):
 - Paralleli standard: $34^{\circ}02'N$ e $35^{\circ}28'N$
 - Meridiano centrale: $118^{\circ}W$
 - $k_0 = 1.0$ ai paralleli standard

Capitolo 8: Calcolo di Distanze e Aree su Ellissoide

8.1 Il Problema Geodetico Diretto e Inverso

Due problemi fondamentali in geodesia:

Problema Diretto: Dati un punto $P_1(\varphi_1, \lambda_1)$, un azimut α_{12} e una distanza s , trovare il punto $P_2(\varphi_2, \lambda_2)$ e l'azimut di ritorno α_{21} .

Problema Inverso: Dati due punti $P_1(\varphi_1, \lambda_1)$ e $P_2(\varphi_2, \lambda_2)$, trovare la distanza geodetica s e gli azimut α_{12} e α_{21} .

Questi problemi sono risolti calcolando la geodetica (curva di minima distanza) sull'ellissoide.

8.2 Geodetiche sull'Ellissoide

Una geodetica su una superficie è una curva che localmente minimizza la distanza. È l'analogo delle linee rette nel piano e dei cerchi massimi sulla sfera.

8.2.1 Equazioni Differenziali delle Geodetiche

Le geodetiche sono soluzioni delle equazioni di Eulero-Lagrange per il funzionale lunghezza:

$$L = \int_{t_1}^{t_2} \sqrt{g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}} dt$$

Questo porta alle equazioni delle geodetiche:

$$\frac{d^2 x^k}{dt^2} + \Gamma^k_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt} = 0$$

Per l'ellissoide in coordinate geografiche, queste sono equazioni differenziali ordinarie del secondo ordine, non risolubili in forma chiusa.

8.2.2 Teorema di Clairaut

Teorema 8.1 (Clairaut): Per una geodetica su una superficie di rivoluzione, il prodotto della distanza dall'asse di rotazione per il seno dell'angolo tra la geodetica e il meridiano è costante lungo la geodetica.

Matematicamente, se α è l'azimut della geodetica e ρ la distanza dall'asse:

$$\rho \sin \alpha = \text{costante}$$

Per l'ellissoide, $\rho = N(\varphi) \cos \varphi$, quindi:

$$N(\varphi) \cos \varphi \sin \alpha = \text{costante} = N(\varphi_0) \cos \varphi_0 \sin \alpha_0$$

dove φ_0 è la latitudine del punto più vicino all'equatore sulla geodetica.

Conseguenze:

1. Una geodetica che parte dall'equatore con $\alpha_0 = 90^\circ$ rimane all'equatore
2. Una geodetica non può attraversare un polo ($\alpha \rightarrow 0$ prima di raggiungerlo)
3. Esiste una latitudine massima $\varphi_{\max}$ dove $\alpha = 90^\circ$

8.3 Formula di Vincenty

Le formule di Vincenty (1975) forniscono soluzioni accurate per i problemi geodetico diretto e inverso.

8.3.1 Problema Inverso di Vincenty

Dati $P_1(\varphi_1, \lambda_1)$ e $P_2(\varphi_2, \lambda_2)$, trovare $s, \alpha_{12}, \alpha_{21}$.

Algoritmo iterativo:

1. Calcolare la latitudine ridotta:

$$U_1 = \arctan[(1-f) \tan \varphi_1]$$

$$U_2 = \arctan[(1-f) \tan \varphi_2]$$

2. Differenza di longitudine:

$$L = \lambda_2 - \lambda_1$$

3. Iterare fino a convergenza (iniziando con $\lambda = L$):

$$\sin \sigma = \sqrt{(\cos U_2 \sin \lambda)^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda)^2}$$

$$\cos \sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos \lambda$$

$$\sigma = \arctan(\sin \sigma, \cos \sigma)$$

$$\sin \alpha = (\cos U_1 \cos U_2 \sin \lambda) / \sin \sigma$$

$$\cos^2 \alpha = 1 - \sin^2 \alpha$$

$$\cos 2\sigma_\square = \cos \sigma - 2 \sin U_1 \sin U_2 / \cos^2 \alpha$$

$$C = (f/16) \cos^2 \alpha [4 + f(4 - 3 \cos^2 \alpha)]$$

$$\lambda_{\text{nuovo}} = L + (1-C)f \sin \alpha \{ \sigma + C \sin \sigma [\cos 2\sigma_\square + C \cos \sigma (-1 + 2 \cos^2 2\sigma_\square)] \}$$

4. Dopo convergenza (tipicamente 2-3 iterazioni):

$$u^2 = \cos^2 \alpha \cdot (a^2 - b^2) / b^2$$

$$A = 1 + (u^2/16384) \{ 4096 + u^2 [-768 + u^2 (320 - 175u^2)] \}$$

$$B = (u^2/1024) \{ 256 + u^2 [-128 + u^2 (74 - 47u^2)] \}$$

$$\Delta \sigma = B \sin \sigma \{ \cos 2\sigma_\square + (B/4) [\cos \sigma (-1 + 2 \cos^2 2\sigma_\square) - (B/6) \cos 2\sigma_\square (-3 + 4 \sin^2 \sigma) (-3 + 4 \cos^2 2\sigma_\square)] \}$$

$$s = b \cdot A (\sigma - \Delta \sigma)$$

$$\alpha_1 = \arctan[(\cos U_2 \sin \lambda) / (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda)]$$

$$\alpha_2 = \arctan[(\cos U_1 \sin \lambda) / (-\sin U_1 \cos U_2 + \cos U_1 \sin U_2 \cos \lambda)]$$

Precisione: Accuratezza di 0.5 mm su distanze fino a 20,000 km.

Casi singolari:

- Punti antipodali: $\lambda = \pi$, algoritmo non converge (infiniti percorsi geodetici di uguale lunghezza)
- Punti quasi antipodali: convergenza lenta, richiedono trattamento speciale

8.3.2 Problema Diretto di Vincenty

Dati $P_1(\varphi_1, \lambda_1)$, α_1 e s , trovare $P_2(\varphi_2, \lambda_2)$ e α_2 .

Algoritmo:

1. Calcolare:

$$\begin{aligned}
U_1 &= \arctan[(1-f) \tan \varphi_1] \\
\sigma_1 &= \arctan[\tan U_1 / \cos \alpha_1] \\
\sin \alpha &= \cos U_1 \sin \alpha_1 \\
u^2 &= \cos^2 \alpha \cdot (a^2 - b^2) / b^2 \\
A &= 1 + (u^2 / 16384) \{4096 + u^2 [-768 + u^2 (320 - 175u^2)]\} \\
B &= (u^2 / 1024) \{256 + u^2 [-128 + u^2 (74 - 47u^2)]\}
\end{aligned}$$

2. Iterare partendo da $\sigma = s / (b \cdot A)$:

$$\begin{aligned}
2\sigma_{\square} &= 2\sigma_1 + \sigma \\
\Delta\sigma &= B \sin \sigma \{...\} \text{ [formula come sopra]} \\
\sigma_{\text{nuovo}} &= s / (b \cdot A) + \Delta\sigma
\end{aligned}$$

3. Dopo convergenza:

$$\begin{aligned}
\varphi_2 &= \arctan[(\sin U_1 \cos \sigma + \cos U_1 \sin \sigma \cos \alpha_1) / ((1-f) \sqrt{(\sin^2 \alpha + (\sin U_1 \sin \sigma - \cos U_1 \cos \sigma \cos \alpha_1)^2))}] \\
\lambda &= \arctan[(\sin \sigma \sin \alpha_1) / (\cos U_1 \cos \sigma - \sin U_1 \sin \sigma \cos \alpha_1)] \\
C &= (f/16) \cos^2 \alpha [4 + f(4 - 3 \cos^2 \alpha)] \\
L &= \lambda - (1-C)f \sin \alpha \{\sigma + C \sin \sigma [\cos 2\sigma_{\square} + C \cos \sigma (-1 + 2 \cos^2 2\sigma_{\square})]\} \\
\lambda_2 &= \lambda_1 + L \\
\alpha_2 &= \arctan[\sin \alpha / (-\sin U_1 \sin \sigma + \cos U_1 \cos \sigma \cos \alpha_1)]
\end{aligned}$$

8.4 Formula di Karney

Thomas Karney (2013) ha sviluppato algoritmi ancora più accurati e robusti, che gestiscono meglio i casi limite.

Vantaggi rispetto a Vincenty:

1. Convergenza garantita per qualsiasi coppia di punti (inclusi antipodali)
2. Accuratezza mantenuta per linee geodetiche molto lunghe
3. Gestione corretta delle geodetiche che attraversano i poli
4. Precisione numerica: errori < 15 nm (nanometri!)

Implementazione: La libreria GeographicLib fornisce implementazioni di riferimento in C++, Python, JavaScript, ecc.

8.5 Calcolo di Aree su Ellissoide

8.5.1 Formula dell'Area di un Poligono Geodetico

Per un poligono i cui lati sono geodetiche, l'area può essere calcolata usando il teorema di Girard generalizzato.

Teorema 8.2 (Area di poligono sferico): L'area di un poligono sferico con n lati è:

$$A = R^2 \cdot E$$

dove E è l'eccesso sferico:

$$E = \sum_{i=1}^n \alpha_i - (n-2)\pi$$

e α_i sono gli angoli interni del poligono.

Per l'ellissoide, la formula è più complessa perché la curvatura non è costante.

8.5.2 Formula di Bessel per Aree Geodetiche

Per un poligono geodetico sull'ellissoide con vertici (φ_i, λ_i) , $i = 1, \dots, n$:

$$A = \frac{1}{2} ab \sum_{i=1}^n (\lambda_{i+1} - \lambda_i)(\sin \varphi_i + \sin \varphi_{i+1}) + \text{correzioni}$$

Le correzioni tengono conto della curvatura non uniforme dell'ellissoide.

Formula più accurata (metodo delle trapezi sulla sfera authalica):

1. Convertire latitudini geodetiche in latitudini authaliche β_i
2. Calcolare l'area sulla sfera authalica (raggio R_A):

$$A = R_A^2 \sum_{i=1}^n (\lambda_{i+1} - \lambda_i) [\sin \beta_i + \sin \beta_{i+1}] / 2$$

dove: $R_A = \sqrt{a^2/2 + b^2/2 \cdot \operatorname{arctanh}(e)/e} \approx 6371007.2 \text{ m}$ (WGS84)

Precisione: Per poligoni con lati $< 100 \text{ km}$, errore $< 0.01\%$.

8.5.3 Algoritmo di Karney per Aree

L'algoritmo più accurato per aree usa l'integrazione della forma differenziale areale lungo il contorno:

$$A = \oint (x dy - y dx)$$

trasformata in coordinate geodetiche. La formula finale coinvolge:

1. Somma degli azimut lungo i lati del poligono
2. Correzioni per la curvatura dell'ellissoide
3. Termini fino all'ordine e^6

Precisione: Accuratezza $< 1 \text{ m}^2$ per poligoni di qualsiasi dimensione.

Capitolo 9: Teoria Avanzata delle Distorsioni

9.1 Tensore Metrico e Distorsione

La distorsione di una proiezione è completamente caratterizzata dal modo in cui il tensore metrico della superficie viene trasformato.

9.1.1 Pullback del Tensore Metrico

Data una proiezione $\varphi: S \rightarrow \mathbb{R}^2$, il tensore metrico del piano viene "tirato indietro" sulla superficie:

$$g_{ij}^{\text{superficie}} = \frac{\partial x^k}{\partial u^i} \frac{\partial x^l}{\partial u^j} \delta_{kl}$$

dove δ è il tensore metrico euclideo del piano.

La differenza tra questo e il tensore metrico intrinseco della superficie misura la distorsione.

9.1.2 Autovalori del Tensore di Distorsione

Il tensore di distorsione D ha autovalori h^2 e k^2 (fattori di scala principali al quadrato). Gli autovettori corrispondono alle direzioni principali.

Invarianti del tensore:

1. Traccia: $\text{tr}(D) = h^2 + k^2$ (somma degli autovalori)
2. Determinante: $\det(D) = h^2 k^2 = \sigma^2$ (distorsione areale)

Interpretazione geometrica:

- $\text{tr}(D)$ misura la distorsione lineare media
- $\det(D)$ misura la distorsione areale
- $\text{tr}(D) - 2\sqrt{\det(D)}$ misura la "non-conformità"

9.2 Indici di Distorsione Globali

Per confrontare proiezioni su regioni finite, definiamo integrali di distorsione.

9.2.1 Distorsione di Tissot Integrata

Indice di Tissot: $I_T = \iint_D \sqrt{(h-k)^2 + (hk-1)^2} \, dA$

Misura la deviazione da una isometria ($h = k = 1$).

9.2.2 Indice di Distorsione di Airy

Formula generale: $I_A = \iint_D (h^2 + k^2 - 2) \, dA$

Per piccole distorsioni, equivalente all'indice di Tissot.

Normalizzazione: Dividiamo per l'area per ottenere la distorsione media: $\bar{I}_A = I_A / \text{Area}(D)$

9.2.3 Indice di Peters (distorsione areale)

$$I_P = \iint_D |h_k - 1| \, dA$$

Zero solo per proiezioni equivalenti.

9.2.4 Indice di Uniformità Angolare

Misura quanto gli angoli sono distorti:

$$I_\omega = \iint_D \omega_{\max}(x,y) \, dA$$

dove $\omega_{\max}$ è la massima distorsione angolare in ogni punto.

9.3 Ottimizzazione di Proiezioni

9.3.1 Problema Variazionale

Trovare la proiezione ottimale minimizzando un funzionale di distorsione:

$$\min_{\phi} \iint_D F(h, k, \nabla h, \nabla k) \, dA$$

soggetta a vincoli (ad esempio, conformità o equivalenza).

Teorema 9.1: Per proiezioni conformi, il minimo di $\iint (h^2 + k^2) \, dA = 2 \iint h^2 \, dA$ si ottiene quando h soddisfa l'equazione di Laplace:

$$\nabla^2 h = 0$$

in coordinate conformalmente equivalenti.

9.3.2 Proiezioni a Minima Errore per Regioni Specifiche

Per una regione D data, la proiezione conforme ottimale risolve:

$$\Delta \ln \mu = 0 \quad \text{in } D \quad \mu = \text{costante sul bordo}$$

Questo è un problema di Dirichlet che può essere risolto numericamente con metodi agli elementi finiti o differenze finite.

Applicazione pratica: Sistemi cartografici nazionali (es. Swiss Oblique Mercator, nuovi sistemi per paesi specifici).

9.4 Visualizzazione delle Distorsioni

9.4.1 Indicatrici di Tissot

Le ellissi di Tissot possono essere disegnate su una mappa per visualizzare la distorsione locale.

Parametri dell'ellisse in un punto:

- Semiasse maggiore: $a = (h+k)/2 + \sqrt{[(h-k)^2/4]}$
- Semiasse minore: $b = (h+k)/2 - \sqrt{[(h-k)^2/4]}$
- Orientazione: $\theta = (1/2)\arctan[2F/(E-G)]$ per coordinate non principali

9.4.2 Mappe di Calore della Distorsione

Rappresentazioni a colori di:

- Fattore di scala: $\mu = \sqrt{hk}$
- Distorsione angolare: $\omega_{\max}$
- Distorsione areale: $\sigma = hk$

Scala di colori tipica:

- Verde: distorsione < 5%
- Giallo: distorsione 5-15%
- Arancione: distorsione 15-30%
- Rosso: distorsione > 30%

Capitolo 10: Implementazione Numerica

10.1 Algoritmi per Proiezioni

10.1.1 Precisione Numerica

Le proiezioni cartografiche richiedono attenzione alla precisione numerica:

1. **Precision floating-point:** IEEE 754 double precision (64 bit) fornisce ~15-16 cifre decimali
2. **Errori di arrotondamento:** Accumulo in iterazioni e somme
3. **Cancellazione catastrofica:** Sottrazione di numeri quasi uguali

Esempio critico: Calcolo di $\sin(x)$ per $x \approx 0$

- $\sin(x) \approx x - x^3/6 + \dots$ per piccolo x
- Usare sviluppo in serie invece di funzione standard per $|x| < 10^{-8}$

10.1.2 Serie di Potenze Ottimizzate

Le formule di proiezione sono spesso espresse come serie di potenze:

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

Schema di Horner per valutazione efficiente: $f(x) = a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + xa_n)\dots))$

Riduce le moltiplicazioni da $n(n+1)/2$ a n .

Esempio (Mercatore Trasversa, coordinate ξ e η):

```
def evaluate_series(coeffs, x):
    result = coeffs[-1]
    for coeff in reversed(coeffs[:-1]):
        result = coeff + x * result
    return result
```

10.1.3 Funzioni Trigonometriche Speciali

Riduzione dell'argomento: Per calcoli trigonometrici accurati:

1. Ridurre l'angolo al primo quadrante: $\theta \rightarrow \theta \bmod (\pi/2)$
2. Usare identità trigonometriche per casi speciali
3. Applicare simmetrie per recuperare il valore originale

Funzioni iperboliche: sinh, cosh, tanh sono critiche per coordinate conforme:

$$\begin{aligned}\sinh(x) &= (e^x - e^{-x})/2 \\ \cosh(x) &= (e^x + e^{-x})/2 \\ \tanh(x) &= (e^{2x} - 1)/(e^{2x} + 1)\end{aligned}$$

Per $|x| < 1$, usare serie:

$$\sinh(x) \approx x + x^3/6 + x^5/120 + \dots$$

10.2 Librerie Software Standard

10.2.1 PROJ

La libreria PROJ (precedentemente PROJ.4) è lo standard de facto per trasformazioni di coordinate.

Caratteristiche:

- Oltre 100 proiezioni implementate
- Trasformazioni tra datum
- Supporto per griglie di trasformazione (NTv2, etc.)
- API in C con binding per Python, JavaScript, ecc.

Esempio di uso (sintassi PROJ):

```
+proj=utm +zone=33 +ellps=WGS84 +datum=WGS84 +units=m +no_defs
```

Transformation pipelines (PROJ 5+):

```
+proj=pipeline
```

```
+step +proj=axisswap +order=2,1
+step +proj=unitconvert +xy_in=deg +xy_out=rad
+step +proj=utm +zone=33 +ellps=WGS84
```

10.2.2 GeographicLib

Libreria di Charles Karney per calcoli geodetici di alta precisione.

Componenti principali:

- Geodesic: problemi diretto e inverso con precisione nanometrica
- TransverseMercator: serie di Kruger fino all'ordine 8
- PolarStereographic, UTMUPS
- Geoid: interpolazione di modelli geoidali

Precisione:

- Distanze geodetiche: errore < 15 nanometri
- Aree: errore < 1 m²
- Convergenza garantita per punti antipodali

10.2.3 Pyproj (Python)

Wrapper Python per PROJ, con interfaccia moderna e conveniente.

Esempio:

```
from pyproj import Proj, transform

# Definire proiezioni
utm33 = Proj(proj='utm', zone=33, ellps='WGS84')
wgs84 = Proj(proj='latlong', datum='WGS84')

# Trasformare coordinate
lon, lat = 12.5, 41.9 # Roma
x, y = transform(wgs84, utm33, lon, lat)
```

10.3 Validazione e Testing

10.3.1 Dataset di Test Standard

NGS (National Geodetic Survey) Test Data:

- Coppie di coordinate note in multipli datum
- Distanze geodetiche calcolate con alta precisione
- Usate per validare implementazioni

Esempio di test case:

P1: $\varphi = 55^{\circ}45'00.00000''\text{N}$, $\lambda = 37^{\circ}37'00.00000''\text{E}$

P2: $\varphi = 39^{\circ}36'41.34209''\text{N}$, $\lambda = 48^{\circ}22'31.10844''\text{E}$

Distanza geodetica: 2,000,000.000 m (esatto)

10.3.2 Casi Limite da Testare

1. **Punti ai poli:** $\varphi = \pm 90^{\circ}$
2. **Attraversamento dell'antimeridiano:** λ vicino a $\pm 180^{\circ}$
3. **Punti antipodali:** massima distanza possibile
4. **Distanze molto brevi:** < 1 m
5. **Azimut vicino a 0° o 180° :** problemi di simmetria

10.3.3 Metriche di Accuratezza

Per confrontare implementazioni:

1. **Errore RMS** (Root Mean Square): $RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i^{\text{computed}} - x_i^{\text{reference}})^2}$
2. **Errore massimo:** $E_{\text{max}} = \max_i |x_i^{\text{computed}} - x_i^{\text{reference}}|$
3. **Percentuale entro tolleranza:** $P_{\varepsilon} = \frac{\#\{i: |x_i^{\text{computed}} - x_i^{\text{reference}}| < \varepsilon\}}{N} \times 100\%$

Tolleranze tipiche:

- Navigazione generale: $\varepsilon = 1\text{-}10$ m
- Rilievi topografici: $\varepsilon = 0.01\text{-}0.1$ m
- Geodesia di precisione: $\varepsilon = 1\text{-}10$ mm
- Metrologia geodetica: $\varepsilon < 1$ mm

10.3.4 Test di Regressione e Validazione Continua

Nella pratica dello sviluppo di software cartografico, è essenziale implementare una suite completa di test automatizzati che verifichino costantemente la correttezza delle implementazioni.

Framework di testing strutturato:

Un sistema di test robusto deve includere diversi livelli di validazione:

1. **Test unitari per funzioni elementari:** Ogni funzione matematica di base (calcolo di $N(\varphi)$, $M(\varphi)$, conversioni trigonometriche, ecc.) deve essere testata individualmente

con valori noti.

2. **Test di integrazione per trasformazioni complete:** Le pipeline di trasformazione complesse devono essere verificate end-to-end con dataset di riferimento certificati.
3. **Test di proprietà matematiche:** Verifica che le proprietà fondamentali siano rispettate (ad esempio, per proiezioni conformi, verificare che $h = k$ ovunque con tolleranza numerica appropriata).

Implementazione di test di invarianza:

Le trasformazioni di coordinate devono rispettare determinate invarianze. Ad esempio, per una trasformazione T e la sua inversa T^{-1} :

Per ogni punto P :

$$P' = T(P)$$

$$P'' = T^{-1}(P')$$

$$\text{assert } \|P - P''\| < \varepsilon$$

Dove ε è una tolleranza appropriata (tipicamente 10^{-6} m per coordinate metriche, 10^{-9} rad per coordinate angolari).

Test di continuità differenziale:

Per proiezioni che devono essere C^1 o C^2 , implementare test numerici delle derivate:

$$f'(x) \approx [f(x+h) - f(x-h)]/(2h)$$

con h piccolo (tipicamente 10^{-6}). Confrontare con la derivata analitica se disponibile, o verificare almeno la continuità calcolando la derivata in punti vicini e assicurandosi che la differenza sia piccola.

Dataset di benchmark standardizzati:

Organizzazioni come NGS (National Geodetic Survey) e EPSG forniscono dataset standardizzati:

- **NGS PC_SOFTWARE Test Data:** Include coordinate in multipli datum con trasformazioni certificate
- **EPSG Guidance Note 7-2:** Centinaia di esempi di trasformazione con risultati attesi
- **IERS Technical Notes:** Dati di riferimento per ITRF

Un esempio concreto dal dataset NGS:

Test Case: GREENWICH_CLASSIC

Punto input (NAD27):

$$\begin{aligned}\varphi &= 38^\circ 55' 17.36659'' \text{ N} \\ \lambda &= 77^\circ 02' 43.82947'' \text{ W} \\ h &= 0.000 \text{ m}\end{aligned}$$

Punto output atteso (NAD83):

$$\begin{aligned}\varphi &= 38^\circ 55' 17.32252'' \text{ N} \\ \lambda &= 77^\circ 02' 43.98814'' \text{ W} \\ h &= 1.660 \text{ m}\end{aligned}$$

Tolleranza:

$$\begin{aligned}\Delta\varphi, \Delta\lambda &< 0.00001'' \\ \Delta h &< 0.001 \text{ m}\end{aligned}$$

10.4 Ottimizzazione Computazionale Avanzata

10.4.1 Lookup Tables e Pre-computazione

Per operazioni frequenti su domini limitati, le lookup tables (LUT) offrono accelerazioni significative sacrificando memoria.

Costruzione di LUT per funzioni trascendenti:

Consideriamo il calcolo di $N(\varphi) = a/\sqrt{1 - e^2 \sin^2 \varphi}$, usato intensivamente nelle conversioni geografiche $\leftrightarrow$ cartesiane.

Strategia:

1. Pre-calcolare $N(\varphi_i)$ per $\varphi_i = i \cdot \Delta\varphi$, $i = 0, 1, \dots, n$ dove $\Delta\varphi = \pi/(2n)$
2. Per φ arbitrario, trovare i tale che $\varphi_i \leq \varphi < \varphi_{i+1}$
3. Interpolare linearmente o cubicamente

Interpolazione lineare:

$$\begin{aligned}i &= \text{floor}(\varphi/\Delta\varphi) \\ t &= (\varphi - \varphi_i)/\Delta\varphi \\ N(\varphi) &\approx (1-t) \cdot N(\varphi_i) + t \cdot N(\varphi_{i+1})\end{aligned}$$

Errore di interpolazione lineare:

Per una funzione $f(x)$ con $|f''(x)| \leq M$, l'errore massimo di interpolazione lineare su intervallo $[a, b]$ è:

$$|\epsilon_{\text{inter}}| \leq M(b-a)^2/8$$

Per $N(\varphi)$ su WGS84:

- $|N''(\varphi)| \leq 0.003 \text{ a}^{-1}$ (massimo ai poli)
- Con $\Delta\varphi = 0.01 \text{ rad} \approx 0.57^\circ$, errore $\leq 4 \times 10^{-8} \text{ m}$

Questo è ampiamente sufficiente per la maggior parte delle applicazioni.

Trade-off memoria-velocità:

Per $n = 9001$ punti (risoluzione 0.01°), la LUT richiede:

- $9001 \times 8 \text{ bytes} \approx 72 \text{ KB}$ (double precision)
- Speedup tipico: 3-5× rispetto a calcolo diretto

Per applicazioni critiche, si possono usare LUT gerarchiche:

- LUT grossolana ($\Delta\varphi = 1^\circ$) per prima approssimazione
- Interpolazione quadratica per raffinamento
- Calcolo diretto solo se precisione richiesta > capacità LUT

10.4.2 Vettorizzazione SIMD

Le istruzioni SIMD (Single Instruction Multiple Data) permettono di processare multipli dati simultaneamente.

Esempio: Conversione batch di coordinate geografiche → UTM

Invece di processare N punti sequenzialmente:

for i in range(N):

$x[i], y[i] = \text{geographic_to_utm}(\text{lat}[i], \text{lon}[i])$

Vettorizziamo usando NumPy (che sfrutta BLAS vettorizzato):

```
import numpy as np
```

```
# lat, lon sono array NumPy di dimensione N
```

```
cos_lat = np.cos(lat)
```

```
sin_lat = np.sin(lat)
```

```
N_array = a / np.sqrt(1 - e2 * sin_lat**2)
```

```
# ... resto dei calcoli vettorizzati
```

Speedup ottenibile:

- CPU moderna (AVX-512): fino a 8 double simultaneamente
- Speedup teorico: 8×
- Speedup pratico: 3-6× (overhead di memoria e branch)

Ottimizzazioni specifiche per architetture:

Intel MKL e OpenBLAS forniscono implementazioni ottimizzate di operazioni vettoriali e matriciali. Per trasformazioni Helmert applicate a milioni di punti:

Trasformazione vettorizzata efficiente

$X_{\text{new}} = \text{translation} + (1 + \text{scale}) * (\text{rotation_matrix} @ X_{\text{old}})$

Usando librerie ottimizzate, questa operazione sfrutta:

- Fused multiply-add (FMA) instructions
- Cache blocking per dati grandi
- Multi-threading automatico

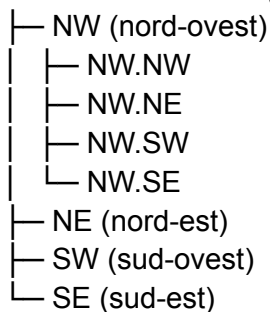
10.4.3 Algoritmi Adattivi e Multi-Risoluzione

Per dati che coprono regioni vaste con variazioni di densità, algoritmi adattivi sono essenziali.

Quad-tree per gestione spaziale:

Struttura gerarchica che suddivide ricorsivamente lo spazio 2D:

Nodo radice: intera regione



Criterio di suddivisione: Un nodo si divide se:

- Contiene più di N punti (tipicamente $N = 100-1000$)
- La profondità massima non è raggiunta (tipicamente 10-15 livelli)

Query spaziale efficiente:

Per trovare tutti i punti entro distanza r da un punto P :

1. Iniziare dalla radice
2. Per ogni nodo, calcolare distanza minima da P al bounding box del nodo
3. Se $\text{distanza_minima} > r$, scartare intero sottoalbero
4. Altrimenti, ricorrere nei figli
5. Ai nodi foglia, testare distanza effettiva per ogni punto

Complessità:

- Costruzione: $O(N \log N)$ medio, $O(N^2)$ peggiore caso
- Query: $O(\log N + k)$ dove k è numero risultati

10.5 Gestione dell'Errore Numerico in Calcoli Critici**10.5.1 Analisi della Stabilità Numerica**

Alcune formule cartografiche sono intrinsecamente instabili numericamente in determinati regimi.

Esempio critico: Calcolo della latitudine da coordinate cartesiane

Il metodo iterativo standard:

$$p = \sqrt{X^2 + Y^2}$$

$$\varphi_{i+1} = \arctan[(Z + e^2 N(\varphi_i) \sin \varphi_i) / p]$$

diventa instabile vicino ai poli dove $p \rightarrow 0$.

Analisi del numero di condizione:

Il numero di condizione κ misura quanto errori nei dati di input amplificano l'errore nell'output:

$$\kappa = \|J\| \cdot \|x\| / \|f(x)\|$$

dove J è la matrice Jacobiana della trasformazione.

Vicino ai poli ($p \approx 0$):

$$\partial \varphi / \partial X = -X \sin \varphi / (p^2 \cos \varphi) \rightarrow \infty \text{ quando } p \rightarrow 0$$

Il numero di condizione diventa arbitrariamente grande, rendendo il problema ill-conditioned.

Soluzione: Formulazione alternativa stabile

Usare coordinate alternative vicino ai poli. Ad esempio, la formulazione di Vermeille evita divisioni per p piccolo usando un cambio di variabili che regolarizza il problema.

10.5.2 Aritmetica Estesa e Compensazione di Kahan

Per somme di molti termini piccoli, l'algoritmo di Kahan (compensated summation) riduce drasticamente l'errore di arrotondamento.

Problema: Calcolare $S = \sum_{i=1}^n x_i$ dove n è grande e alcuni x_i sono piccoli.

Somma ingenua:

```
s = 0.0
for x in data:
    s = s + x
```

Errore accumulato: $O(n\epsilon|S|)$ dove $\epsilon \approx 10^{-16}$ (epsilon macchina per double).

Algoritmo di Kahan:

```
s = 0.0
c = 0.0 # Compensazione
for x in data:
    y = x - c
    t = s + y
    c = (t - s) - y
    s = t
```

Errore accumulato: $O(\epsilon|S|)$ (indipendente da n !)

Applicazione in cartografia:

Calcolo di aree di poligoni con molti vertici usando la formula del trapezio:

$$A = (1/2) |\sum_i (x_i y_{i+1} - x_{i+1} y_i)|$$

Per poligoni con 10^6 vertici, la compensazione di Kahan è essenziale per mantenere precisione centimetrica.

10.5.3 Arithmetic Interval (Aritmetica degli Intervalli)

Per valutare rigorosamente i limiti dell'errore numerico, l'aritmetica degli intervalli rappresenta ogni quantità come $[a^-, a^+]$ dove il valore vero è garantito essere nell'intervallo.

Operazioni base:

$$\begin{aligned} [a^-, a^+] + [b^-, b^+] &= [a^- + b^-, a^+ + b^+] \\ [a^-, a^+] - [b^-, b^+] &= [a^- - b^+, a^+ - b^-] \\ [a^-, a^+] \times [b^-, b^+] &= [\min(a^-b^-, a^-b^+, a^+b^-, a^+b^+), \\ &\quad \max(a^-b^-, a^-b^+, a^+b^-, a^+b^+)] \end{aligned}$$

Problema: L'ampiezza degli intervalli tende a crescere esponenzialmente con la profondità del calcolo (dependency problem).

Soluzione parziale: Affine arithmetic, che traccia correlazioni lineari tra variabili.

Applicazione: Validazione rigorosa di librerie cartografiche. L'aritmetica degli intervalli può certificare che, dati input con errore $\leq \epsilon$, l'output ha errore $\leq \delta(\epsilon)$ calcolabile.

Capitolo 11: Teoria delle Proiezioni Policentriche e Multi-Scala

11.1 Proiezioni con Discontinuità Intenzionali

Le proiezioni tradizionali sono continue su tutto il dominio. Tuttavia, per minimizzare la distorsione su regioni molto estese o con forme particolari, è spesso vantaggioso introdurre discontinuità controllate.

11.1.1 Proiezioni Interrotte

Definizione 11.1: Una proiezione interrotta è una proiezione che viene applicata separatamente a più lobi o fusi, con interruzioni lungo meridiani selezionati.

Esempio canonico - Proiezione Sinusoidale Interrotta di Goode:

La proiezione sinusoidale standard ha forte distorsione lontano dal meridiano centrale. Goode (1925) propose di usare 6 lobi separati:

Lobi per i continenti:

- Lobo 1: America del Nord e Sud (meridiano centrale: 100°W)
- Lobo 2: Europa e Africa (meridiano centrale: 20°E)
- Lobo 3: Asia (meridiano centrale: 140°E)
- Lobo 4-6: Regioni pacifiche e oceaniche

Matematica delle interruzioni: Ogni lobo usa la formula sinusoidale standard: $x = R\lambda' \cos \varphi$, $y = R\varphi$

dove $\lambda' = \lambda - \lambda_0$ è la longitudine relativa al meridiano centrale del lobo.

Funzione di selezione del lobo: Per un punto (λ, φ) , definiamo: $\text{lobo}(\lambda, \varphi) = \arg\min_i D_i(\lambda, \varphi)$

dove D_i è la distorsione nel lobo i .

Proprietà:

1. Equivalenza preservata in ogni lobo
2. Distorsione massima ridotta drasticamente (da ~300% a ~40%)
3. Discontinuità lungo meridiani di interruzione
4. Non utilizzabile per navigazione (percorsi attraversano interruzioni)

11.1.2 Teoria delle Proiezioni Composite

Definizione 11.2: Una proiezione composita è definita a pezzi su diverse regioni, con possibili transizioni continue o discontinue.

Classificazione:

1. **Composizione discontinua:** Interruzioni evidenti (es. Goode)
2. **Composizione con blending:** Transizione graduale tra proiezioni
3. **Composizione gerarchica:** Diverse proiezioni a scale diverse

Teorema 11.3: Non esiste una proiezione composita finita che sia simultaneamente continua, conforme ed equivalente su tutto il globo.

Dimostrazione: Segue dal Theor4. **XGM2019e:**

- Modello più recente (2019)
- Combinazione di dati satellitari e terrestri
- Massima risoluzione e accuratezza attuale

Selezione Level of Detail (LOD): Basata su:

- Zoom level dell'utente
- Importanza features (strade principali vs viuzze)
- Densità di dati (urbano vs rurale)

11.2 Proiezioni Multi-Scala e Adaptive Refinement

11.2.1 Piramidi di Proiezione

Per rappresentazioni cartografiche che coprono scale multiple (come web maps), si utilizza una gerarchia di proiezioni a risoluzioni diverse.

Web Mercator Tile System (WMTS):

Il sistema più diffuso per mappe online (Google Maps, OpenStreetMap, ecc.) divide il mondo in tiles (tessere) organizzate in livelli di zoom.

Struttura matematica:

Livello di zoom z :

- Il mondo è diviso in $2^z \times 2^z$ tiles
- Ogni tile ha dimensione fissa (tipicamente 256×256 pixel)
- Risoluzione effettiva: $r(z) = C/(2^z \cdot 256)$ dove C è la circonferenza equatoriale

Coordinate tile:

Data una posizione geografica (λ, φ) , il tile (t_x, t_y) al livello z è:

$$n = 2^z$$

$$t_x = \text{floor}(n \cdot (\lambda + \pi)/(2\pi))$$

$$t_y = \text{floor}(n \cdot (1 - \ln(\tan \varphi + \sec \varphi)/\pi)/2)$$

Nota: La formula per t_y deriva dalla proiezione di Mercatore.

Coordinate pixel all'interno di un tile:

$$p_x = \text{floor}(256 \cdot (n \cdot (\lambda + \pi)/(2\pi) - t_x))$$

$$p_y = \text{floor}(256 \cdot (n \cdot (1 - \ln(\tan \varphi + \sec \varphi)/\pi)/2 - t_y))$$

Livelli di zoom e scala:

La relazione tra zoom level z e scala cartografica (a latitudine φ) è:

$$\text{Scala} = C \cdot \cos \varphi / (2^z \cdot 256 \cdot \text{dpi}/0.0254)$$

Assumendo dpi = 96 (standard web):

$z = 0$	→	Scala 1:500,000,000	(mondo intero)
$z = 5$	→	Scala 1:16,000,000	(continente)
$z = 10$	→	Scala 1:500,000	(regione)
$z = 15$	→	Scala 1:16,000	(città)
$z = 18$	→	Scala 1:2,000	(quartiere)
$z = 20$	→	Scala 1:500	(edificio)

11.2.2 Adaptive Mesh Refinement (AMR) per Proiezioni

Per regioni con geometria complessa o requisiti di accuratezza variabili, l'AMR permette di allocare risoluzione computazionale solo dove necessario.

Principio: Iniziare con una griglia grossolana, poi raffinare iterativamente celle dove l'errore supera una soglia.

Implementazione per proiezioni:

1. **Griglia iniziale:** Dividere il dominio (sferico o ellissoidale) in celle grossolane
2. **Stima errore:** Per ogni cella, stimare l'errore di proiezione (distorsione lineare, angolare, areale)
3. **Raffinamento:** Suddividere celle con errore $> \varepsilon_{\text{soglia}}$
4. **Iterazione:** Ripetere fino a convergenza

Stima dell'errore per cella:

Per una cella con vertici (λ_i, φ_i) , $i = 1, 2, 3, 4$:

1. Proiettare i vertici
2. Calcolare il fattore di scala h, k ai vertici
3. Errore stimato: $E = \max(|h - 1|, |k - 1|)$ oppure $E = |hk - 1|$ (per equivalenza)

Criterio di raffinamento:

Una cella viene raffinata se:

- $E > \varepsilon_{\text{soglia}}$ (tipicamente 0.01-0.05)
- Oppure se la variazione di E entro la cella è grande (per catturare gradienti)

Vantaggi:

- Memoria e calcolo concentrati dove servono
- Possibile ottenere accuratezza uniforme desiderata su domini complessi
- Naturalmente parallelizzabile

Esempio di applicazione: Proiezione ottimale per la Norvegia (forma molto allungata nord-sud con fiordi). Una proiezione conica standard ha distorsioni elevate. Con AMR:

- Costa e fiordi: raffinamento alto
- Mare aperto: raffinamento basso
- Risultato: proiezione pseudo-conica con mesh irregolare che minimizza distorsione globale

11.3 Proiezioni Conformi Generalizzate e Trasformazioni di Schwarz-Christoffel

11.3.1 Teoria delle Mappe Conformi nel Piano Complesso

Le proiezioni conformi della sfera (o ellissoide) sul piano sono intimamente connesse alla teoria delle funzioni analitiche complesse.

Richiamo: Una funzione $f: \mathbb{C} \rightarrow \mathbb{C}$ è analitica (olomorfa) se soddisfa le equazioni di Cauchy-Riemann:

$$\begin{aligned}\partial u / \partial x &= \partial v / \partial y \\ \partial u / \partial y &= -\partial v / \partial x\end{aligned}$$

dove $f(z) = u(x,y) + iv(x,y)$.

Proprietà fondamentale: Le funzioni analitiche sono conformi (preservano angoli) ovunque la derivata $f'(z) \neq 0$.

Applicazione alla cartografia:

Usando coordinate isometriche (λ, ψ) dove ψ è la latitudine conforme, l'ellissoide diventa localmente "piatto" nel senso della metrica conforme:

$$ds^2 = h^2(\lambda, \psi)(d\lambda^2 + d\psi^2)$$

Una proiezione conforme è quindi una funzione analitica:

$$w = f(z) \text{ dove } z = \lambda + i\psi$$

e $w = x + iy$ sono le coordinate cartesiane.

11.3.2 Trasformazione di Schwarz-Christoffel

La trasformazione di Schwarz-Christoffel mappa conformemente l'asse reale (o cerchio unitario) su un poligono arbitrario.

Forma generale (mappa asse reale $\rightarrow$ poligono):

$$w = f(z) = A + C \int \prod_{i=1}^n (\zeta - z_i)^{\alpha_i/\pi - 1} d\zeta$$

dove:

- z_i sono punti sull'asse reale
- α_i sono gli angoli esterni del poligono ai vertici
- A, C sono costanti di traslazione e scala

Proprietà notevole: Questa formula mappa l'interno del semipiano superiore conformemente sull'interno del poligono.

Applicazione cartografica:

Per proiettare conformemente una regione con confine poligonale (ad esempio, uno stato o provincia con confini rettilinei in coordinate geografiche):

1. Identificare i vertici del confine in coordinate isometriche (λ_i, ψ_i)
2. Calcolare i parametri z_i e α_i
3. Applicare la trasformazione di Schwarz-Christoffel
4. Il risultato è una proiezione conforme ottimale per quella regione

Difficoltà computazionale:

Il calcolo dei parametri z_i richiede la soluzione di un sistema non lineare di equazioni trascendenti. Questo è un problema numerico delicato che richiede:

- Buona inizializzazione (usando approssimazioni analitiche)
- Algoritmi iterativi robusti (Newton-Raphson multi-dimensionale)
- Regolarizzazione per evitare singolarità

Software specializzato: Il pacchetto SC Toolbox (MATLAB/Python) implementa algoritmi efficienti per Schwarz-Christoffel.

11.3.3 Proiezioni Conformi Doppie

Per regioni che non sono ben approssimate da forme standard (cerchi, ellissi, rettangoli), le "doppie proiezioni" conformi offrono flessibilità.

Principio:

1. Prima proiezione conforme: Sfera $\rightarrow$ Dominio intermedio D (es. disco, rettangolo)
2. Seconda proiezione conforme: D $\rightarrow$ Piano finale

Esempio classico: Proiezione conforme ottimale per l'Islanda

L'Islanda ha forma irregolare che non si adatta bene a proiezioni standard. La proiezione ufficiale islandese (Lambert 1993) usa:

1. Proiezione stereografica obliqua centrata sull'Islanda $\rightarrow$ disco
2. Trasformazione conforme del disco ottimizzata per minimizzare distorsione

Il risultato ha distorsione massima $< 0.5\%$ sull'isola, eccezionalmente basso.

Costruzione della trasformazione ottimale:

Per una regione R, cercare la funzione analitica f che minimizza:

$$I = \iint_R (|f'(z)|^2 - 1)^2 dA$$

soggetta a vincoli al bordo (ad esempio, che punti specifici siano mappati in posizioni desiderate).

Questo porta a un problema variazionale la cui soluzione soddisfa un'equazione differenziale complessa. Numericamente, si discretizza usando elementi finiti o differenze finite nel dominio complesso.

11.4 Proiezioni per Corpi Celesti Non-Sferici

11.4.1 Mappe di Corpi Irregolari

Per oggetti celesti piccoli (asteroidi, comete, satelliti irregolari), la forma è troppo irregolare per essere approssimata da ellissoidi.

Approccio basato su mesh poliédrica:

1. La superficie è rappresentata da una mesh triangolare (tipicamente da dati LiDAR o fotogrammetrici)
2. Ogni triangolo è mappato individualmente sul piano

3. Si cerca di minimizzare la distorsione globalmente

Proiezione di Gauss per triangoli:

Dato un triangolo T con vertici p_1, p_2, p_3 :

1. Scegliere un sistema di coordinate locale tangente in p_1
2. Proiettare i vertici usando sviluppo di Taylor della metrica
3. Questo produce un triangolo piano approssimato

Distorsione in proiezione di mesh:

Per ogni triangolo, misurare:

- Distorsione lineare: rapporto tra lati più lungo e più corto
- Distorsione angolare: differenza massima tra angoli originali e proiettati
- Distorsione areale: $|Area_proiettata/Area_originale - 1|$

Ottimizzazione globale:

Trovare una disposizione dei triangoli nel piano che minimizzi la distorsione totale è un problema di ottimizzazione combinatoria difficile (simile al problema del travelling salesman in 2D).

Euristiche usate:

- Iniziare con una proiezione greedy (espandere triangolo per triangolo minimizzando localmente)
- Applicare algoritmi di rilassamento (spostare vertici per ridurre stress)
- Simulated annealing o algoritmi genetici per ottimizzazione globale

Software: Il pacchetto SPICE della NASA include strumenti per mappe di corpi irregolari.

11.4.2 Mappe Planetarie a Diversa Gravità

Per pianeti con rotazione rapida o bassa massa (quindi schiacciamento elevato), l'ellissoide standard potrebbe non essere adeguato.

Esempio: Saturno

Saturno ha schiacciamento $f \approx 0.098$ (circa 10%!), molto maggiore della Terra ($f \approx 0.003$).

Per rappresentazioni accurate:

1. Usare ellissoidi con e^2 corrispondente
2. Le formule standard si applicano, ma i termini di ordine superiore in e diventano significativi
3. Serie che su Terra convergono in 2-3 termini richiedono 5-10 termini

Parametri approssimati di Saturno:

- a (equatoriale) $\approx 60,268$ km
- b (polare) $\approx 54,364$ km
- $f = (a-b)/a \approx 0.098$
- $e^2 = 2f - f^2 \approx 0.186$

Formule di proiezione modificate:

Per Mercatore su Saturno:

$$\psi(\varphi) = \operatorname{arctanh}(\sin \varphi) - e \cdot \operatorname{arctanh}(e \sin \varphi)$$

Con $e = 0.431$, il secondo termine è significativo anche per φ moderati:

- A $\varphi = 30^\circ$: secondo termine ≈ 0.22 (22% del primo!)
- Sulla Terra ($e = 0.082$): secondo termine ≈ 0.04 (4% del primo)

11.4.3 Proiezioni per Superfici Non-Convesse

Alcuni satelliti o asteroidi hanno forme "a patata" con concavità significative.

Problema: La nozione stessa di "proiezione" diventa ambigua se la normale alla superficie non è unica o se raggi dalla superficie si intersecano.

Approccio: Proiezione da scheletro curvilineo

1. Calcolare lo "scheletro" (medial axis) della forma
2. Usare lo scheletro come "asse di proiezione"
3. Ogni punto sulla superficie è proiettato lungo la normale allo scheletro più vicino

Alternativa: Unwrapping basato su curvatura principale

1. Identificare le "creste" (ridge lines) di massima curvatura
2. "Aprire" la superficie lungo queste creste
3. Sviluppare le patch risultanti sul piano minimizzando distorsione

Questo è analogo a sbucciare un'arancia: inevitabilmente ci sono discontinuità, ma posizionate strategicamente.

Capitolo 12: Trasformazioni di Coordinate e Conversioni tra Sistemi

12.1 Catene di Trasformazione

Nella pratica geodetica moderna, raramente si converte direttamente tra due sistemi arbitrari. Invece, si usa una catena di trasformazioni elementari.

12.1.1 Pipeline di Trasformazione Standard

Esempio: Conversione da coordinate geografiche NAD27 a coordinate piane State Plane NAD83.

Pipeline completa:

1. $(\lambda, \phi, h)_{\text{NAD27}} \rightarrow (X, Y, Z)_{\text{NAD27}}$ (conversione a cartesiane geocentriche)
2. $(X, Y, Z)_{\text{NAD27}} \rightarrow (X, Y, Z)_{\text{NAD83}}$ (trasformazione Helmert)
3. $(X, Y, Z)_{\text{NAD83}} \rightarrow (\lambda, \phi, h)_{\text{NAD83}}$ (conversione a geografiche)
4. $(\lambda, \phi)_{\text{NAD83}} \rightarrow (E, N)_{\text{StatePlane}}$ (proiezione cartografica)

Notazione PROJ pipeline:

```
+proj=pipeline
+step +proj=cart +ellps=clrk66 # NAD27 → cartesiane
+step +proj=helmert +x=-9 +y=160 +z=176 # trasformazione datum
+step +inv +proj=cart +ellps=GRS80 # cartesiane → NAD83
+step +proj=lcc +lat_1=34.03333333 +lat_2=35.46666667 +lat_0=33.5
+lon_0=-118 +x_0=2000000 +y_0=500000 +ellps=GRS80 # proiezione
```

12.1.2 Algebra delle Trasformazioni

Le trasformazioni di coordinate formano un gruppoide (categoria con tutti i morfismi invertibili).

Composizione: Se $T_1: A \rightarrow B$ e $T_2: B \rightarrow C$, allora $T_2 \circ T_1: A \rightarrow C$

Inversa: Ogni trasformazione $T: A \rightarrow B$ ha un'inversa $T^{-1}: B \rightarrow A$

Proprietà:

1. Associatività: $(T_3 \circ T_2) \circ T_1 = T_3 \circ (T_2 \circ T_1)$
2. Identità: $T \circ \text{id} = \text{id} \circ T = T$
3. Inversa: $T \circ T^{-1} = T^{-1} \circ T = \text{id}$

Grafo di trasformazione: Rappresentiamo i sistemi di coordinate come nodi e le trasformazioni come archi diretti.

Algoritmo per trovare la trasformazione ottimale $A \rightarrow B$:

1. Costruire il grafo di tutti i sistemi e trasformazioni disponibili
2. Trovare il percorso più breve (Dijkstra) pesato per accuratezza
3. Comporre le trasformazioni lungo il percorso

Implementazione in PROJ: Il registro EPSG contiene >6000 sistemi di coordinate e >10000 trasformazioni. PROJ costruisce automaticamente pipeline ottimali.

12.2 Conversioni Geografiche ↔ Cartesiane Geocentriche

12.2.1 Geografiche → Cartesiane (Algoritmo Diretto)

Date coordinate geografiche (λ , φ , h) su ellissoide con parametri (a , e^2):

$$X = (N(\varphi) + h) \cos \varphi \cos \lambda \quad Y = (N(\varphi) + h) \cos \varphi \sin \lambda \quad Z = (N(\varphi)(1 - e^2) + h) \sin \varphi$$

dove $N(\varphi) = a/\sqrt{1 - e^2 \sin^2 \varphi}$ è il raggio di curvatura del primo verticale.

Complessità: $O(1)$, calcolo diretto.

Precisione: Limitata solo dalla precisione floating-point ($\sim 10^{-15} \times |\text{coordinate}|$).

12.2.2 Cartesiane $\rightarrow$ Geografiche (Problema Inverso)

Date coordinate cartesiane (X , Y , Z), trovare (λ , φ , h).

Longitudine (triviale): $\lambda = \arctan2(Y, X)$

Latitudine e altezza (non triviali): Richiedono iterazione.

Metodo 1: Iterazione di Newton-Raphson

Partiamo con una stima iniziale φ_0 :

- Calcolare $p = \sqrt{X^2 + Y^2}$
- $\varphi_0 = \arctan(Z/p)$ (approssimazione sferica)

Iterare: $N = a/\sqrt{1 - e^2 \sin^2 \varphi_n}$ $\varphi_{n+1} = \arctan[(Z + e^2 N \sin \varphi_n)/p]$

fino a convergenza (tipicamente 3-4 iterazioni per precisione millimetrica).

Poi: $h = p/\cos \varphi - N$

Metodo 2: Formula di Bowring (1976)

Più efficiente, convergenza garantita.

Passo 1: Calcolare parametri ausiliari $p = \sqrt{X^2 + Y^2}$ $e'^2 = (a^2 - b^2)/b^2$ $E = b^2/a$

Passo 2: Prima approssimazione $F = 54b^2Z^2$ $G = p^2 + (1 - e^2)Z^2 - e^2(a^2 - b^2)$ $c = (e^4 F p^2)/G^3$ $s = \sqrt[3]{1 + c + \sqrt{c^2 + 2c}}$ $P = F/(3(s + 1/s + 1)^2 G^2)$ $Q = \sqrt{1 + 2e^4 P}$

Passo 3: Calcolo finale $r_0 = -(Pe^2 p)/(1 + Q) + \sqrt{((a^2/2)(1 + 1/Q) - (P(1 - e^2)Z^2)/(Q(1 + Q)) - P p^2/2)}$
 $U = \sqrt{((p - e^2 r_0)^2 + Z^2)}$ $V = \sqrt{((p - e^2 r_0)^2 + (1 - e^2)Z^2)}$

$\varphi = \arctan[(a/b)(Z/V)]$ $h = U(1 - b^2/(aV))$

Metodo 3: Formula di Vermeille (2002)

Ancora più robusta, senza iterazione.

$$L = e^2/2 \quad L2 = L^2 \quad m = (p/a)^2 \quad n = ((1-e^2)/e^2)(Z/b)^2 \quad i = -(2L2 + m + n)/2 \quad k = L2(L2 - m - n) \quad q = (m + n - 4L2)^3/216 + L2 mn$$

Calcolare la radice appropriata e procedere con trasformazioni algebriche complesse (50+ operazioni).

Confronto dei metodi:

- Newton-Raphson: Semplice ma richiede iterazione
- Bowring: Buon compromesso, ~20 operazioni
- Vermeille: Nessuna iterazione ma complesso, ~60 operazioni

Casi singolari:

- Al centro ($X = Y = Z = 0$): latitudine indefinita, $h = -N$ (qualunque φ)
- Sull'asse polare ($X = Y = 0$): longitudine indefinita
- A grande distanza ($|r| \gg a$): precisione degradata

12.3 Trasformazioni tra Ellissoidi

12.3.1 Trasformazione di Molodensky Standard

Per piccole variazioni di datum senza rotazioni significative.

Input: ($\lambda_1, \varphi_1, h_1$) su ellissoide 1 Output: ($\lambda_2, \varphi_2, h_2$) su ellissoide 2

Parametri:

- $\Delta a = a_2 - a_1$
- $\Delta f = f_2 - f_1$
- $\Delta X, \Delta Y, \Delta Z$ (traslazioni origine)

Calcoli ausiliari: $M = a_1(1 - e_1^2)/(1 - e_1^2 \sin^2 \varphi_1)^{3/2}$ $N = a_1/(1 - e_1^2 \sin^2 \varphi_1)^{1/2}$

Formule: $\Delta \varphi = [-\Delta X \sin \varphi_1 \cos \lambda_1 - \Delta Y \sin \varphi_1 \sin \lambda_1 + \Delta Z \cos \varphi_1 + (a_1 \Delta f + f_1 \Delta a)(M/a_1 + N/a_1) \sin \varphi_1 \cos \varphi_1] / (M + h_1)$

$$\Delta \lambda = (-\Delta X \sin \lambda_1 + \Delta Y \cos \lambda_1) / [(N + h_1) \cos \varphi_1]$$

$$\Delta h = \Delta X \cos \varphi_1 \cos \lambda_1 + \Delta Y \cos \varphi_1 \sin \lambda_1 + \Delta Z \sin \varphi_1 - (a_1 \Delta f + f_1 \Delta a)(N/a_1) \sin^2 \varphi_1 + \Delta a$$

$$\text{Risultato: } \lambda_2 = \lambda_1 + \Delta \lambda \quad \varphi_2 = \varphi_1 + \Delta \varphi \quad h_2 = h_1 + \Delta h$$

Accuratezza: ~5-10 m per trasformazioni tra datum moderni

Limitazioni:

- Valida solo per piccole rotazioni ($< 1''$)
- Accuratezza degrada per $h > 10$ km

12.3.2 Trasformazione di Helmert Estesa

Per trasformazioni di alta precisione.

Forma a 14 parametri:

- 3 traslazioni: (t_x, t_y, t_z)
- 3 rotazioni: (r_x, r_y, r_z)
- 1 scala: s
- 3 velocità di traslazione: ($\dot{t}_x, \dot{t}_y, \dot{t}_z$)
- 3 velocità di rotazione: ($\dot{r}_x, \dot{r}_y, \dot{r}_z$)
- 1 velocità di scala: $\dot{s}$

Per epoca t : $t(t) = t_0 + \dot{t}(t - t_0)$ $R(t) = R_0 + \dot{R}(t - t_0)$ $s(t) = s_0 + \dot{s}(t - t_0)$

Trasformazione completa: $X_2(t) = t(t) + [1 + s(t)]R(t) X_1(t)$

Applicazione: Trasformazioni tra diverse realizzazioni di ITRF, che includono moto delle placche tettoniche.

Esempio ITRF2014 $\rightarrow$ ITRF2008:

Epoca riferimento: 2010.0

$t_x = 1.6$ mm, $\dot{t}_x = 0.0$ mm/anno

$t_y = 1.9$ mm, $\dot{t}_y = 0.0$ mm/anno

$t_z = -2.4$ mm, $\dot{t}_z = -0.1$ mm/anno

$s = -0.02$ ppb, $\dot{s} = +0.03$ ppb/anno

$r_x = 0.0$ mas, $\dot{r}_x = 0.0$ mas/anno

...

12.4 Griglie di Trasformazione

12.4.1 Concetto di Griglia NTv2

Per trasformazioni datum che non possono essere modellate accuratamente con parametri globali (Helmert, Molodensky), si usano griglie di correzione.

Formato NTv2 (National Transformation version 2):

- Griglia regolare di punti in latitudine/longitudine
- Ad ogni punto: ($\Delta\phi, \Delta\lambda$) correzioni
- Interpolazione bilineare tra punti griglia

Struttura file NTv2:

Header generale (informazioni griglia)

Sub-griglia 1 (regione A, risoluzione alta)

Array di shift: $\Delta\phi[i,j]$, $\Delta\lambda[i,j]$

Sub-griglia 2 (regione B, risoluzione media)

...
 Sub-griglia parent (intera nazione, risoluzione bassa)
 ...

Algoritmo di applicazione:

1. Identificare la sub-griglia più fine che contiene il punto
2. Trovare i 4 punti griglia circostanti
3. Interpolare bilinearmente le correzioni
4. Applicare: $\varphi_2 = \varphi_1 + \Delta\varphi$, $\lambda_2 = \lambda_1 + \Delta\lambda$

Interpolazione bilineare: Date le correzioni ai 4 angoli:

- $(\varphi_i, \lambda_j): \Delta\varphi_{11}, \Delta\lambda_{11}$
- $(\varphi_i, \lambda_{j+1}): \Delta\varphi_{12}, \Delta\lambda_{12}$
- $(\varphi_{i+1}, \lambda_j): \Delta\varphi_{21}, \Delta\lambda_{21}$
- $(\varphi_{i+1}, \lambda_{j+1}): \Delta\varphi_{22}, \Delta\lambda_{22}$

Parametri normalizzati: $s = (\varphi - \varphi_i) / (\varphi_{i+1} - \varphi_i)$ $t = (\lambda - \lambda_j) / (\lambda_{j+1} - \lambda_j)$

Interpolazione: $\Delta\varphi = (1-s)(1-t)\Delta\varphi_{11} + (1-s)t\Delta\varphi_{12} + s(1-t)\Delta\varphi_{21} + st\Delta\varphi_{22}$
 $\Delta\lambda = (1-s)(1-t)\Delta\lambda_{11} + (1-s)t\Delta\lambda_{12} + s(1-t)\Delta\lambda_{21} + st\Delta\lambda_{22}$

Esempi di griglie NTV2:

- NTV2_0.gsb (Canada): NAD27 $\rightarrow$ NAD83
- NADCON (USA): NAD27 $\rightarrow$ NAD83
- RGF93 (Francia): NTF $\rightarrow$ RGF93
- OSTN15 (UK): OSGB36 $\rightarrow$ ETRS89

Accuratezza: Tipicamente 0.01-0.5 m, molto superiore a trasformazioni parametriche.

12.4.2 Griglie di Ondulazione Geoidale

Simili a NTV2 ma per conversione altezze:

h (ellissoidale) = H (ortometrica) + N (ondulazione geoide)

Formato: File raster con valori N a griglia regolare

Esempi:

- EGM96: 15' $\times$ 15' (risoluzione ~28 km) Ogni misurazione ha incertezza. Come si propaga attraverso trasformazioni?

Legge di propagazione dell'errore (prima approssimazione lineare):

Se $y = f(x)$ e x ha matrice di covarianza Σ_x , allora y ha matrice di covarianza:

$$\Sigma_y = J \Sigma_x J^T$$

dove $J = \partial f / \partial x$ è la matrice Jacobiana.

Esempio: Conversione geografiche $\rightarrow$ cartesiane

$$x = [\lambda, \varphi, h]^T \rightarrow y = [X, Y, Z]^T$$

Jacobiana:

$$J = \begin{bmatrix} \partial X / \partial \lambda & \partial X / \partial \varphi & \partial X / \partial h \\ \partial Y / \partial \lambda & \partial Y / \partial \varphi & \partial Y / \partial h \\ \partial Z / \partial \lambda & \partial Z / \partial \varphi & \partial Z / \partial h \end{bmatrix}$$

Elementi espliciti: $\partial X / \partial \lambda = -(N + h) \cos \varphi \sin \lambda$ $\partial X / \partial \varphi = -(N + h) \sin \varphi \cos \lambda + (\partial N / \partial \varphi) \cos \varphi \cos \lambda$ $\partial X / \partial h = \cos \varphi \cos \lambda \dots$ (9 derivate parziali totali)

Incertezze tipiche:

- GPS standard: $\sigma_\lambda = \sigma_\varphi \approx 5 \text{ m}$, $\sigma_h \approx 10 \text{ m}$
- GPS differenziale: $\sigma_\lambda = \sigma_\varphi \approx 0.01\text{-}0.1 \text{ m}$, $\sigma_h \approx 0.02\text{-}0.2 \text{ m}$
- GPS RTK: $\sigma_\lambda = \sigma_\varphi = \sigma_h \approx 0.01\text{-}0.02 \text{ m}$

12.5.2 Ellissi ed Ellissoidi di Errore

L'incertezza in 2D è visualizzata come ellisse di confidenza.

Data matrice di covarianza 2×2 :

$$\Sigma = \begin{bmatrix} \sigma_{x^2} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{y^2} \end{bmatrix}$$

Autovalori λ_1, λ_2 (con $\lambda_1 \geq \lambda_2$) danno i semiassi dell'ellisse: $a = k\sqrt{\lambda_1}$ $b = k\sqrt{\lambda_2}$

dove k dipende dal livello di confidenza:

- $k = 1.000$: $\sim 39.3\%$ (1σ)
- $k = 1.960$: 95.0%
- $k = 2.448$: 99.0%
- $k = 2.576$: 99.0%

Orientazione dell'ellisse: $\theta = (1/2) \arctan(2\sigma_{xy}/(\sigma_{x^2} - \sigma_{y^2}))$

In 3D, l'ellissoide di errore ha 3 semiassi corrispondenti ai 3 autovalori di Σ .

12.5.3 Combinazione di Misurazioni Multiple

Principio: Più misurazioni indipendenti di una stessa quantità possono essere combinate per ridurre l'incertezza.

Media pesata: Date n misurazioni x_i con incertezze σ_i :

$$\hat{x} = (\sum w_i x_i) / (\sum w_i)$$

dove i pesi sono $w_i = 1/\sigma_i^2$

Incetezza della media: $\sigma_{\hat{x}} = 1/\sqrt{(\sum w_i)} = 1/\sqrt{(\sum 1/\sigma_i^2)}$

Riduzione incertezza: Con n misurazioni di uguale qualità σ : $\sigma_{\hat{x}} = \sigma/\sqrt{n}$

Metodo dei minimi quadrati: Per sovradeterminazione (più equazioni che incognite), minimizzare:

$$\chi^2 = \sum [(osservato_i - calcolato_i)/\sigma_i]^2$$

$$\text{Soluzione: } \hat{x} = (A^T W A)^{-1} A^T W b$$

dove $W = \text{diag}(1/\sigma_i^2)$ è la matrice dei pesi.

12.6 Griglie di Trasformazione Avanzate

12.6.1 Formato NTv2: Struttura Interna Dettagliata

Il formato NTv2 (National Transformation version 2) è uno standard binario per griglie di trasformazione. Comprendiamo la sua struttura interna per implementazioni corrette.

Struttura gerarchica del file:

[Header Generale]

- Identificativo formato: "NUM_OREC", "NUM_SREC", "NUM_FILE"
- Ellissoide sorgente e destinazione
- Metadata: data creazione, agenzia, ecc.

[Sub-griglia 1 - Alta risoluzione]

Header sub-griglia:

- Nome univoco (es. "TORONTO_DETAIL")
- Limiti geografici: lat_min, lat_max, lon_min, lon_max
- Spaziatura: dlat, dlon (in arc-seconds)
- Dimensioni: nrows, ncols
- Sub-griglia parent (se esiste)

Dati griglia:

- Array 2D di shift: per ogni nodo (i,j):
 - lat_shift (arc-seconds)
 - lon_shift (arc-seconds)

- lat_accuracy (meters)
- lon_accuracy (meters)

[Sub-griglia 2 - Risoluzione media]

...

[Sub-griglia N - Risoluzione bassa (copertura nazionale)]

...

Algoritmo di lettura e applicazione:

```
def apply_ntv2_shift(lat, lon, ntv2_grid):
    # 1. Trovare sub-griglia appropriata
    subgrid = find_finetest_subgrid(lat, lon, ntv2_grid)

    if subgrid is None:
        raise ValueError("Punto fuori da tutte le griglie")

    # 2. Calcolare indici griglia
    i_float = (lat - subgrid.lat_min) / subgrid.dlat
    j_float = (lon - subgrid.lon_min) / subgrid.dlon

    i = int(floor(i_float))
    j = int(floor(j_float))

    # Verificare bounds
    if not (0 <= i < subgrid.nrows-1 and 0 <= j < subgrid.ncols-1):
        # Punto vicino al bordo, usare griglia parent
        return apply_ntv2_shift(lat, lon, subgrid.parent)

    # 3. Interpolazione bilineare
    s = i_float - i # frazione verticale
    t = j_float - j # frazione orizzontale

    lat_shift = (1-s)*(1-t)*subgrid.lat_shift[i,j] + \
        (1-s)*t*subgrid.lat_shift[i,j+1] + \
        s*(1-t)*subgrid.lat_shift[i+1,j] + \
        s*t*subgrid.lat_shift[i+1,j+1]

    lon_shift = (1-s)*(1-t)*subgrid.lon_shift[i,j] + \
        (1-s)*t*subgrid.lon_shift[i,j+1] + \
        s*(1-t)*subgrid.lon_shift[i+1,j] + \
        s*t*subgrid.lon_shift[i+1,j+1]

    # 4. Convertire da arc-seconds a gradi decimali
```

```
lat_shift_deg = lat_shift / 3600.0
lon_shift_deg = lon_shift / 3600.0
```

```
return lat + lat_shift_deg, lon + lon_shift_deg
```

Interpolazione bicubica per maggiore precisione:

Per applicazioni che richiedono accuratezza sub-centimetrica, l'interpolazione bilineare potrebbe non essere sufficiente. Si usa allora interpolazione bicubica che utilizza 16 punti (griglia 4×4) invece di 4.

La formula bicubica generale per una funzione $f(x,y)$ è:

$$f(x,y) \approx \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

dove i coefficienti a_{ij} sono determinati imponendo che la superficie interpoli i valori e le derivate ai 4 punti d'angolo.

Errore di interpolazione:

Per una griglia con spaziatura h :

- Interpolazione lineare: errore $O(h^2)$
- Interpolazione bicubica: errore $O(h^4)$

Per $h = 1'$ (arc-minute ≈ 1.85 km):

- Lineare: errore ~ 3 -5 cm
- Bicubica: errore ~ 0.1 -0.3 mm

12.6.2 Griglie di Deformazione Tempo-Varianti

Le placche tettoniche si muovono costantemente. Per trasformazioni tra datum con epoche diverse, servono griglie che includono velocità di deformazione.

Modello di deformazione:

$$P(t) = P(t_0) + V \cdot (t - t_0) + D(x,y) \cdot (t - t_0)$$

dove:

- $P(t_0)$: posizione all'epoca di riferimento
- V : velocità media (da modello placche)
- $D(x,y)$: campo di deformazione locale (da griglia)

Formato HTDP (Horizontal Time Dependent Positioning):

Sviluppato da NGS per gli USA, include:

- Velocità orizzontali (v_e , v_n) in ogni nodo griglia
- Modelli di eventi sismici (jump istantanei)
- Deformazione post-sismica (rilassamento esponenziale)

Esempio: Terremoto di San Andreas

Un evento sismico produce:

1. Spostamento co-sismico (istantaneo): Δu_0
2. Rilassamento post-sismico:

$$\Delta u(t) = \Delta u_0 + A(1 - e^{-(t/\tau)})$$

dove $\tau \approx 10\text{-}20$ anni è la costante di tempo del mantello.

12.6.3 Griglie Geoidali ad Alta Risoluzione

Le griglie geoidali moderne raggiungono risoluzioni spaziali impressionanti.

GEOID18 (USA):

- Risoluzione: $1' \times 1'$ (1.85 km)
- Copertura: USA continentali, Alaska, Hawaii, Porto Rico
- Accuratezza: 1-2 cm (RMS) in aree ben controllate
- Basato su: GRAV-D (airborne gravity), EGM2008, GPS/leveling

Formato di file tipico:

Header:

```
nlat = 1801 (numero righe)
nlon = 3601 (numero colonne)
lat0 = 24.0° (latitudine minima)
lon0 = -125.0° (longitudine minima)
dlat = 0.0166667° (1 arc-minute)
dlon = 0.0166667°
```

Data:

```
N[0,0] N[0,1] ... N[0,3600]    (riga sud)
N[1,0] N[1,1] ... N[1,3600]
...
N[1800,0] ... N[1800,3600]    (riga nord)
```

Ogni valore $N[i,j]$ è l'ondulazione del geoide (tipicamente in metri, con 0.001 m di precisione).

Compressione dei dati:

Una griglia completa del geoide globale a 1' richiede:

- $10801 \times 21601 \approx 233$ milioni di punti
- 4 bytes per punto (float32) ≈ 932 MB
- Non gestibile per applicazioni mobile

Tecniche di compressione:

1. **Compressione differenziale:** Memorizzare differenze invece di valori assoluti

$$d[i,j] = N[i,j] - N[i,j-1]$$

Le differenze sono piccole $\rightarrow$ comprimibili

2. **Wavelets:** Decomposizione in componenti multi-scala
 - Componente a bassa frequenza (trend generale): griglia grossolana
 - Dettagli ad alta frequenza: coefficienti wavelet sparsi
 - Compressione: scartare coefficienti piccoli
3. **Compressione lossy:** PNG con 16-bit quantizzazione
 - Errore introdotto: < 1 mm
 - Fattore di compressione: 10-20×

12.7 Concatenazione Ottimale di Trasformazioni

12.7.1 Grafi di Trasformazione e Algoritmi di Ricerca

In un sistema GIS moderno, ci sono centinaia di sistemi di coordinate. EPSG Registry contiene oltre 6000 CRS (Coordinate Reference Systems).

Rappresentazione come grafo:

- Nodi: CRS (ad es. "EPSG:4326" = WGS84, "EPSG:32633" = WGS84/UTM zone 33N)
- Archi diretti: Trasformazioni disponibili (con metadati: accuratezza, metodo)

Esempio di rete parziale:

WGS84 Geographic (4326)

- └→ WGS84/UTM 33N (32633) [proiezione, accuratezza: 1 mm]
- └→ ETRS89 Geographic (4258) [datum shift, accuratezza: 1 cm]
- └→ NAD83 Geographic (4269) [datum shift, accuratezza: 1 m]

ETRS89 Geographic (4258)

- └→ ETRS89/UTM 33N (25833) [proiezione, accuratezza: 1 mm]
- └→ ETRS89/LAEA (3035) [proiezione, accuratezza: 1 mm]

NAD83 Geographic (4269)

└→ NAD83/State Plane California V (2229) [proiezione]
└→ WGS84 Geographic (4326) [identità approssimata, accuratezza: 1-2 m]

Problema: Dato CRS sorgente S e destinazione D, trovare la sequenza ottimale di trasformazioni.

Algoritmo di Dijkstra pesato per accuratezza:

```
def find_transformation_path(source, target, crs_graph):
    # Inizializzazione
    distances = {source: 0} # accuratezza cumulativa
    previous = {}
    unvisited = set(crs_graph.nodes)

    while unvisited:
        # Nodo con minima distanza (massima accuratezza cumulativa)
        current = min(unvisited, key=lambda n: distances.get(n, float('inf')))

        if current == target:
            break

        if distances.get(current) == float('inf'):
            break # Nessun percorso

        unvisited.remove(current)

        # Esplorare vicini
        for neighbor in crs_graph.neighbors(current):
            edge = crs_graph.get_edge(current, neighbor)

            # Combinare errori in quadratura (indipendenti)
            new_distance = sqrt(distances[current]**2 + edge.accuracy**2)

            if new_distance < distances.get(neighbor, float('inf')):
                distances[neighbor] = new_distance
                previous[neighbor] = (current, edge)

    # Ricostruire percorso
    path = []
    node = target
    while node != source:
        if node not in previous:
            return None # Nessun percorso
        prev_node, transformation = previous[node]
        path.insert(0, (prev_node, node, transformation))
        node = prev_node
```

return path

Combinazione degli errori:

Se due trasformazioni T_1 e T_2 hanno accuratze σ_1 e σ_2 , l'accuratezza della composizione $T_2 \circ T_1$ è:

$$\sigma_{\text{totale}} = \sqrt{(\sigma_1^2 + \sigma_2^2)}$$

assumendo errori indipendenti gaussiani (approssimazione ragionevole per la maggior parte delle trasformazioni geodetiche).

Esempio concreto:

Trasformare da "Italia Roma 1940 / Gauss-Boaga Ovest" (EPSG:3003) a "ETRS89 / UTM 33N" (EPSG:25833):

Percorso ottimale trovato:

1. EPSG:3003 → EPSG:4265 (Roma 1940 geographic) [proiezione inversa, 0.1 m]
2. EPSG:4265 → EPSG:4326 (WGS84 geographic) [datum shift 7-param, 3 m]
3. EPSG:4326 → EPSG:4258 (ETRS89 geographic) [identità moderna, 0.01 m]
4. EPSG:4258 → EPSG:25833 (ETRS89 UTM 33N) [proiezione, 0.001 m]

Accuratezza totale: $\sqrt{(0.1^2 + 3^2 + 0.01^2 + 0.001^2)} \approx 3.0$ m

Percorsi alternativi più accurati:

Se disponibile griglia NTV2 specifica Roma1940 → ETRS89:

1. EPSG:3003 → EPSG:4265 [proiezione inversa, 0.1 m]
2. EPSG:4265 → EPSG:4258 [griglia NTV2, 0.5 m]
3. EPSG:4258 → EPSG:25833 [proiezione, 0.001 m]

Accuratezza totale: $\sqrt{(0.1^2 + 0.5^2 + 0.001^2)} \approx 0.51$ m (miglioramento 6×!)

12.7.2 Caching e Ottimizzazione di Pipeline

Per applicazioni che trasformano milioni di punti tra le stesse coppie di CRS, il caching è essenziale.

Strategia di caching:

```
class TransformationCache:
    def __init__(self, max_size=100):
        self.cache = {}
```

```

self.max_size = max_size
self.access_count = {}

def get_transformation(self, source, target):
    key = (source, target)

    if key in self.cache:
        self.access_count[key] += 1
        return self.cache[key]

    # Calcolare e cachare
    pipeline = find_transformation_path(source, target)
    compiled_transform = compile_pipeline(pipeline)

    if len(self.cache) >= self.max_size:
        # Evictare elemento meno usato (LFU)
        lfu_key = min(self.access_count, key=self.access_count.get)
        del self.cache[lfu_key]
        del self.access_count[lfu_key]

    self.cache[key] = compiled_transform
    self.access_count[key] = 1

    return compiled_transform

```

Compilazione di pipeline:

Una pipeline come:

WGS84 → cartesiane → Helmert → cartesiane → NAD83 → UTM

può essere "compilata" in una funzione unica ottimizzata che:

- Pre-calcola costanti
- Evita conversioni intermedie non necessarie
- Sfrutta vettorizzazione per batch

Ottimizzazione algebrica automatica:

Alcune composizioni di trasformazioni possono essere semplificate:

Esempio:

T_1 : Proiezione $(\lambda, \varphi) \rightarrow (x, y)$

T_2 : Traslazione $(x, y) \rightarrow (x+a, y+b)$

T_3 : Rotazione $(x, y) \rightarrow (x', y')$

T_4 : Scala $(x', y') \rightarrow (sx', sy')$

Invece di applicare 4 trasformazioni separate, comporre in trasformazione affine unica:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} s \cdot \cos \theta & -s \cdot \sin \theta \\ s \cdot \sin \theta & s \cdot \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a' \\ b' \end{bmatrix}$$

dove a' , b' incorporano la traslazione originale trasformata.

Teorema (Composizione di trasformazioni affini):

La composizione di n trasformazioni affini è essa stessa affine:

$$T = T_n \circ T_{n-1} \circ \dots \circ T_1$$

può essere scritta come:

$$T(x) = Ax + b$$

dove A è il prodotto delle matrici e b è calcolato ricorsivamente.

Questo permette di ridurre n operazioni matriciali a 1 sola per ogni punto.

12.8 Trasformazioni Temporalì e Posizionamento 4D

12.8.1 Coordinate 4D: Spazio-Tempo

Nei sistemi geodetici moderni, la posizione è una funzione del tempo:

$$P(t) = (X(t), Y(t), Z(t))$$

Cause di variazione temporale:

1. **Moto delle placche tettoniche:** 2-10 cm/anno (costante)
2. **Deformazione elastica continua:** Accumulo di stress alle faglie
3. **Eventi co-sismici:** Jump istantanei (terremoti)
4. **Rilassamento post-sismico:** Rilassamento viscoso del mantello
5. **Carico glaciale:** Rebound isostatico (mm/anno)
6. **Maree terrestri:** Variazione periodica (10-30 cm, ciclo ~12 ore)
7. **Carico atmosferico:** Pressione atmosferica (mm)
8. **Carico oceanico:** Livello del mare (mm)

Modello completo:

$$P(t) = P_0 + V \cdot (t - t_0) + \sum_i H(t - t_i) \cdot D_i + R(t) + M(t) + \varepsilon(t)$$

dove:

- P_0 : posizione all'epoca di riferimento t_0
- V : velocità secolare (placche)
- $H(t - t_i) \cdot D_i$: funzioni step (Heaviside) per terremoti all'epoca t_i
- $R(t)$: rilassamento post-sismico
- $M(t)$: maree + altri effetti periodici
- $\varepsilon(t)$: rumore

Esempio numerico: Stazione GPS in California

Epoca riferimento: 2010.0

Posizione: $P_0 = (X_0, Y_0, Z_0)$ in ITRF2014

Velocità placca: $V = (-0.012, 0.028, 0.005)$ m/anno

Terremoti principali:

- 2014.5: $D_1 = (-0.15, 0.08, 0.02)$ m
- 2019.8: $D_2 = (-0.05, 0.03, 0.01)$ m

Posizione a 2023.5:

$$t = 2023.5 - 2010.0 = 13.5 \text{ anni}$$

$$\begin{aligned} P(2023.5) &= P_0 + V \cdot 13.5 + D_1 + D_2 \\ &= P_0 + (-0.162, 0.378, 0.0675) + (-0.15, 0.08, 0.02) + (-0.05, 0.03, 0.01) \\ &= P_0 + (-0.362, 0.488, 0.0975) \text{ m} \end{aligned}$$

Spostamento totale: $||\Delta P|| = 0.607$ m in 13.5 anni

12.8.2 Trasformazioni tra Epoche

Per trasformare coordinate da un'epoca a un'altra nello stesso datum:

Trasformazione lineare (prima approssimazione):

$$P(t_2) = P(t_1) + V \cdot (t_2 - t_1)$$

dove V è il campo di velocità, disponibile da:

- Modelli globali: NUVEL-1A, GSRM v2.1, MORVEL
- Griglie locali: da reti CORS dense

Trasformazione completa (con discontinuità):

```

def transform_epoch(P_t1, t1, t2, velocity_grid, earthquake_db):
    # Velocità secolare
    V = velocity_grid.interpolate(P_t1)
    P_t2 = P_t1 + V * (t2 - t1)

    # Applicare eventi sismici nell'intervallo [t1, t2]
    for earthquake in earthquake_db.between(t1, t2):
        if earthquake.affects(P_t1):
            # Calcolare spostamento co-sismico
            D_coseis = earthquake.coseismic_displacement(P_t1)
            P_t2 += D_coseis

            # Aggiungere rilassamento post-sismico
            tau = earthquake.relaxation_time
            t_since = t2 - earthquake.time
            D_relax = earthquake.relaxation_amplitude(P_t1) * (1 - exp(-t_since/tau))
            P_t2 += D_relax

    return P_t2

```

Accuratezza della trasformazione temporale:

Per intervalli temporali:

- $\Delta t < 1$ anno: accuratezza ~ 1 cm (dominata da misura velocità)
- $1 < \Delta t < 10$ anni: accuratezza $\sim 2-5$ cm (incertezza velocità + eventi sismici non modellati)
- $\Delta t > 10$ anni: accuratezza $\sim 5-20$ cm (deformazione non lineare significativa)

12.8.3 Sistemi di Riferimento Semi-Dinamici

Un datum "semi-dinamico" fissa le coordinate a un'epoca di riferimento ma include il modello di velocità.

Esempio: ITRF2014

Ogni stazione ha coordinate all'epoca 2010.0 più velocità:

Stazione ZIMM (Zimmerwald, Svizzera):

$X(2010.0) = 4331297.3480$ m $\dot{X} = -0.01462$ m/anno

$Y(2010.0) = 567555.6390$ m $\dot{Y} = 0.01917$ m/anno

$Z(2010.0) = 4633133.7270$ m $\dot{Z} = 0.01365$ m/anno

Posizione a qualsiasi epoca t :

$$X(t) = X(2010.0) + \dot{X} \cdot (t - 2010.0)$$

$$Y(t) = Y(2010.0) + \dot{Y} \cdot (t - 2010.0)$$

$$Z(t) = Z(2010.0) + \dot{Z} \cdot (t - 2010.0)$$

Datum "Statico" vs "Semi-Dinamico" vs "Dinamico":

1. **Statico (es. NAD83 originale):** Coordinate fisse, non aggiornate. Utile per mappe catastali stabili ma diverge dalla realtà fisica.
2. **Semi-Dinamico (es. ITRF):** Coordinate + velocità. Posizione a qualsiasi epoca calcolabile ma richiede aggiornamento delle coordinate per uso pratico.
3. **Dinamico (teorico):** Coordinate aggiornate in tempo reale. Impraticabile per usi cartografici ma usato in navigazione GNSS real-time.

Trade-off pratico:

Per applicazioni catastali/cartografiche, datum statici sono preferiti (stabilità giuridica). Per applicazioni scientifiche/geodetiche, datum dinamici sono essenziali.

Alcuni paesi stanno transitando a datum semi-dinamici con "plate-frozen" variants:

- Coordinate all'epoca di riferimento
- Velocità sottratta è quella della placca tettonica principale
- Deformazione intra-placca rimane

Capitolo 13: Applicazioni Moderne e Tecnologie Emergenti

13.1 GNSS e Posizionamento Satellitare

13.1.1 Sistemi GNSS Globali

GPS (Stati Uniti):

- Costellazione: 24+ satelliti
- Orbite: MEO (~20,200 km)
- Sistema di riferimento: WGS84
- Frequenze: L1 (1575.42 MHz), L2 (1227.60 MHz), L5 (1176.45 MHz)
- Precisione: 3-10 m (standard), 0.01-0.1 m (differenziale)

GLONASS (Russia):

- Costellazione: 24 satelliti
- Sistema di riferimento: PZ-90 (simile a WGS84)
- Frequenze: FDMA tradizionale, CDMA modernizzato
- Copertura: Ottima alle alte latitudini

Galileo (Europa):

- Costellazione: 24+ satelliti (completamento in corso)
- Sistema di riferimento: GTRF (Galileo Terrestrial Reference Frame)
- Frequenze: E1, E5a, E5b, E6
- Precisione: Servizio commerciale ad alta precisione disponibile

BeiDou (Cina):

- Costellazione: 35+ satelliti (GEO + IGSO + MEO)
- Copertura: Globale (BDS-3)
- Sistema di riferimento: CGCS2000

13.1.2 Posizionamento Preciso (PPP)

Precise Point Positioning: Tecnica che utilizza correzioni orbitali e di clock precise per ottenere accuratezza centimetrica con singolo ricevitore.

Principio:

1. Osservazioni di codice e fase su due frequenze
2. Modelli precisi di orbite satellitari (da IGS)
3. Correzioni di clock satellitare ad alta frequenza
4. Correzioni ionosferiche e troposferiche

Equazione di osservazione (forma linearizzata): $P = \rho + c(dt_r - dt_s) + I + T + \epsilon_P$ $\Phi = \rho + c(dt_r - dt_s) - I + T + \lambda N + \epsilon_\Phi$

dove:

- P: pseudorange (osservazione di codice)
- Φ : phase range (osservazione di fase)
- ρ : distanza geometrica vera
- dt_r, dt_s : errori di clock ricevitore/satellite
- I: ritardo ionosferico
- T: ritardo troposferico
- N: ambiguità intera (fase)
- λ : lunghezza d'onda

Convergenza: PPP richiede 20-30 minuti per convergere a precisione centimetrica (problema: risoluzione ambiguità).

PPP-RTK: Combina PPP con correzioni RTK per convergenza rapida (<1 minuto).

13.1.3 Reti CORS (Continuously Operating Reference Stations)

Concetto: Rete di stazioni GNSS permanenti che forniscono correzioni in tempo reale.

Servizi:

1. **Posizionamento differenziale:** Correzioni da singola stazione base
2. **RTK in rete:** Interpolazione correzioni da multiple stazioni
3. **Post-processing:** Dati storici per elaborazione offline

VRS (Virtual Reference Station):

- Il sistema genera una stazione virtuale vicino all'utente
- Correzioni più accurate (errori spazialmente correlati ridotti)
- Nessuna dipendenza da distanza singola stazione

Algoritmo VRS:

1. Utente invia posizione approssimata
2. Server calcola correzioni alle stazioni reali circostanti
3. Server interpola correzioni alla posizione utente
4. Genera dati come se esistesse stazione fisica lì

13.2 LiDAR e Nuvole di Punti

13.2.1 Acquisizione LiDAR

Principio: Laser scanner misura distanze emettendo impulsi laser e misurando tempo di ritorno.

Equazione fondamentale: $d = ct/2$

dove:

- d: distanza
- c: velocità della luce
- t: tempo di andata-ritorno

LiDAR aereo (Airborne Laser Scanning):

- Altezza volo: 500-5000 m
- Densità punti: 1-50 punti/m²
- Accuratezza verticale: 5-15 cm
- Accuratezza orizzontale: 30-50 cm

Componenti del sistema:

1. Laser scanner (misura distanza + angolo)
2. IMU (Inertial Measurement Unit - orientazione)
3. GNSS (posizione)

Georeferenziazione punto: $P_{\text{ground}} = P_{\text{GNSS}} + R_{\text{IMU}} \cdot R_{\text{scanner}} \cdot d \cdot \hat{u}$

dove:

- P_ground: posizione punto terreno in coordinate globali
- P_GNSS: posizione antenna GNSS
- R_IMU: matrice rotazione da IMU
- R_scanner: rotazione scanner interna
- d: distanza misurata
- $\hat{u}$: vettore unitario direzione laser

13.2.2 Generazione DTM/DSM da LiDAR

DTM (Digital Terrain Model): Superficie terreno nudo **DSM** (Digital Surface Model): Superficie superiore (include vegetazione, edifici)

Pipeline di elaborazione:

1. **Classificazione punti:** Ground/Non-ground
 - Algoritmi: TIN progressivo, morfologia matematica, machine learning
2. **Filtraggio:** Rimozione outliers, rumore
3. **Interpolazione:** Creazione superficie continua
 - Metodi: TIN (Triangulated Irregular Network), IDW (Inverse Distance Weighting), Kriging
4. **Generazione raster:** Conversione a griglia regolare

Algoritmo di classificazione ground (semplificato):

1. Trovare punti più bassi in celle griglia
2. Creare TIN iniziale da questi punti
3. Per ogni punto non classificato:
 - Calcolare distanza e angolo al TIN
 - Se angolo < soglia E distanza < soglia: classificare come ground
4. Aggiornare TIN con nuovi punti ground
5. Iterare fino a convergenza

13.3 Fotogrammetria Digitale e Structure from Motion

13.3.1 Principi di Fotogrammetria

Equazioni di collinearità: Relazione tra punto terreno (X, Y, Z), punto immagine (x, y), e centro proiezione (X₀, Y₀, Z₀):

$$x - x_0 = -f [r_{11}(X-X_0) + r_{12}(Y-Y_0) + r_{13}(Z-Z_0)] / [r_{31}(X-X_0) + r_{32}(Y-Y_0) + r_{33}(Z-Z_0)]$$

$$y - y_0 = -f [r_{21}(X-X_0) + r_{22}(Y-Y_0) + r_{23}(Z-Z_0)] / [r_{31}(X-X_0) + r_{32}(Y-Y_0) + r_{33}(Z-Z_0)]$$

dove:

- f : lunghezza focale
- (x_0, y_0) : punto principale (centro immagine)
- r_{ij} : elementi matrice di rotazione

Stereo fotogrammetria: Usa due immagini sovrapposte per ricostruire 3D.

Parallasse: Differenza posizione punto nelle due immagini fornisce informazione di profondità.

Profondità: $Z = (B \cdot f)/p$

dove:

- B : baseline (distanza tra camere)
- f : focale
- p : parallasse

13.1.2 Structure from Motion (SfM)

Rivoluzione moderna: Ricostruzione 3D automatica da collezione non organizzata di foto.

Pipeline SfM:

1. **Feature detection:** Identificare punti caratteristici (SIFT, SURF, ORB)
2. **Feature matching:** Trovare corrispondenze tra immagini
3. **Bundle adjustment incrementale:**
 - Inizializzare con coppia immagini
 - Aggiungere immagini successive
 - Ottimizzare simultaneamente pose camere e struttura 3D
4. **Dense reconstruction:** Multi-view stereo per nuvola densa

Bundle adjustment: Minimizzare errore di riproiezione:

$$\min \sum_i \sum_j \|p_{ij} - \pi(P_j, C_i)\|^2$$

dove:

- p_{ij} : punto osservato nell'immagine i
- P_j : punto 3D j
- C_i : posa camera i
- π : funzione proiezione

Applicazioni:

- Rilievi archeologici
- Modellazione edifici
- Ortofoto ad alta risoluzione

- Mappe 3D urbane

13.4 Intelligenza Artificiale e Deep Learning in Cartografia

13.4.1 Estrazione Automatica di Features

Reti neurali convoluzionali (CNN) per classificazione immagini satellitari:

Architetture comuni:

- U-Net: Segmentazione semantica (pixel-by-pixel)
- ResNet: Classificazione con skip connections
- Mask R-CNN: Instance segmentation (singoli oggetti)

Applicazioni:

1. **Estrazione edifici:** Rilevamento automatico footprints
2. **Classificazione uso suolo:** 10+ classi (urbano, agricolo, foresta, acqua, ...)
3. **Estrazione rete stradale:** Centerline automatica
4. **Rilevamento cambiamenti:** Confronto multi-temporale

Dataset di training:

- SpaceNet: Immagini satellitari + ground truth edifici, strade
- ImageNet: Pre-training per features generiche
- OpenStreetMap: Labels crowd-sourced

13.4.2 Generazione Automatica Mappe

Map generalization con ML:

- Input: Mappa dettagliata (es. scala 1:10,000)
- Output: Mappa generalizzata (es. scala 1:100,000)
- Training: Coppie di mappe professionalmente generalizzate

Operazioni di generalizzazione:

1. Semplificazione: Ridurre vertici linee
2. Smoothing: Rendere curve più fluide
3. Aggregazione: Unire features vicine
4. Displacement: Spostare per risolvere conflitti
5. Selezione: Omettere features meno importanti

Approccio tradizionale (algoritmi deterministici):

- Douglas-Peucker per semplificazione
- Regole expert-based per displacement
- Risultati inconsistenti, richiedono tuning manuale

Approccio ML- Metrologia geodetica: $\varepsilon < 1 \text{ mm}$

13.5 Cartografia in Realtà Aumentata e Virtuale

13.5.1 Sistemi di Coordinate per AR/VR

La realtà aumentata richiede allineamento preciso tra mondo virtuale e reale, che pone sfide uniche per la cartografia.

Sistema di coordinate AR:

Un'applicazione AR tipica gestisce multipli sistemi:

1. **World Coordinates:** Sistema globale (WGS84 o locale)
2. **Camera Coordinates:** Sistema solidale con la camera del dispositivo
3. **Screen Coordinates:** Coordinate 2D dello schermo
4. **Object Coordinates:** Sistema locale per ogni oggetto virtuale

Catena di trasformazioni:

World $\rightarrow$ Camera: $T_{wc}(t)$ [pose del dispositivo, da GPS + IMU + SLAM]

Camera $\rightarrow$ Screen: P [proiezione prospettica della camera]

Per un punto world P_w , la sua proiezione sullo schermo è:

$$p_{\text{screen}} = P \cdot T_{wc} \cdot P_w$$

Proiezione prospettica:

La matrice di proiezione della camera è:

$$P = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

dove:

- f_x, f_y : lunghezze focali in pixel
- c_x, c_y : coordinate del centro ottico

Trasformazione completa:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = P \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

$$\begin{bmatrix} 1 \end{bmatrix}$$

$$u_{\text{screen}} = u/w$$

$$v_{\text{screen}} = v/w$$

Calibrazione della camera:

Per AR preciso, è essenziale calibrare:

1. Parametri intrinseci: f_x , f_y , c_x , c_y , distorsione radiale k_1 , k_2 , ...
2. Allineamento GPS-IMU-Camera: offset di traslazione e rotazione

Modello di distorsione:

Lenti reali introducono distorsione (specialmente wide-angle su smartphone):

$$x_{\text{distorted}} = x_{\text{ideal}}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{\text{distorted}} = y_{\text{ideal}}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$\text{dove } r^2 = x_{\text{ideal}}^2 + y_{\text{ideal}}^2$$

13.5.2 SLAM (Simultaneous Localization and Mapping) e Cartografia

SLAM è la tecnica di costruire una mappa dell'ambiente mentre simultaneamente si localizza il dispositivo in essa.

SLAM visuale:

Usa immagini da camera per:

1. Estrarre feature (punti salienti, angoli, bordi)
2. Tracciare feature tra frame consecutivi
3. Stimare movimento della camera (visual odometry)
4. Costruire mappa 3D di feature

Formulazione matematica:

Stato del sistema:

$$x = [x_{\text{camera}}, M]$$

dove:

- x_{camera} = (posizione, orientazione) della camera
- $M = \{P_1, P_2, \dots, P_n\}$ posizioni 3D delle feature

Osservazioni: proiezioni 2D delle feature nelle immagini

Extended Kalman Filter (EKF) SLAM:

Predizione:

$$\hat{\mathbf{x}}_t | \mathbf{z}_{t-1} = f(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_t) \text{ [modello di movimento]}$$

$$\mathbf{P}_t | \mathbf{z}_{t-1} = \mathbf{F}_t \mathbf{P}_{t-1} \mathbf{F}_t^T + \mathbf{Q}_t \text{ [covarianza predetta]}$$

Aggiornamento:

$$\mathbf{K}_t = \mathbf{P}_t | \mathbf{z}_{t-1} \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t | \mathbf{z}_{t-1} \mathbf{H}_t^T + \mathbf{R}_t)^{-1} \text{ [guadagno di Kalman]}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t | \mathbf{z}_{t-1} + \mathbf{K}_t (\mathbf{z}_t - h(\hat{\mathbf{x}}_t | \mathbf{z}_{t-1})) \text{ [correzione]}$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_t | \mathbf{z}_{t-1} \text{ [covarianza aggiornata]}$$

Problema di scala in visual SLAM monoculare:

Una camera singola non può determinare la scala assoluta (tutto può essere scalato uniformemente). Soluzioni:

1. Stereo camera: baseline nota fornisce scala
2. Fusione con IMU: accelerometro fornisce scala via integrazione
3. Fusione con GPS: posizioni assolute forniscono scala

13.5.3 Georeferenziazione di Scene 3D

Per integrare oggetti virtuali con dati cartografici reali, serve georeferenzizzare la mappa SLAM.

Allineamento globale:

Dato:

- Mappa SLAM locale $\mathbf{M}_{\text{slam}} = \{\mathbf{P}_1, \dots, \mathbf{P}_n\}$ (coordinate arbitrarie)
- Corrispondenze GPS $\mathbf{P}_{\text{gps}}(t_i)$ a varie epoche

Trovare trasformazione di similarità (7 DoF):

$$\mathbf{P}_{\text{world}} = s \cdot \mathbf{R} \cdot \mathbf{P}_{\text{slam}} + \mathbf{t}$$

che minimizza:

$$\sum_i \|\mathbf{P}_{\text{gps}}(t_i) - (s \cdot \mathbf{R} \cdot \mathbf{P}_{\text{slam}}(t_i) + \mathbf{t})\|^2$$

Algoritmo di allineamento:

1. Selezionare almeno 3 corrispondenze non collineari

2. Calcolare centroidi: c_{slam} , c_{gps}
3. Matrici centrate: $P'_{slam} = P_{slam} - c_{slam}$, $P'_{gps} = P_{gps} - c_{gps}$
4. Matrice di covarianza: $H = \sum P'_{slam,i} \cdot P'_{gps,i}^T$
5. SVD: $H = U \Sigma V^T$
6. Rotazione: $R = VU^T$ (con correzione per $\det(R) = +1$)
7. Scala: $s = \text{tr}(\Sigma) / \text{tr}(P'_{slam} \cdot P'_{slam}^T)$
8. Traslazione: $t = c_{gps} - s \cdot R \cdot c_{slam}$

Precisione dell'allineamento:

Dipende da:

- Accuratezza GPS: tipicamente 3-10 m (smartphone), 0.01-0.1 m (RTK)
- Numero di corrispondenze: più è meglio
- Distribuzione spaziale: corrispondenze ben distribuite danno geometria migliore

Per AR outdoor con GPS smartphone:

- Errore posizionale: 5-15 m
- Sufficiente per "object placement" approssimativo
- Non sufficiente per allineamento preciso con modelli 3D esistenti

Per AR professionale (rilievi, BIM):

- RTK GPS: errore 2-5 cm
- Total station: errore sub-mm
- Allineamento con modelli CAD preciso

13.6 Cartografia Dinamica e Mappe Temporali

13.6.1 Rappresentazione di Fenomeni Temporali

Molti fenomeni geografici evolvono nel tempo: traffico, meteo, inquinamento, uso del suolo, ecc.

Modelli di dati spazio-temporali:

1. **Snapshot:** Sequenza di mappe statiche a epoche discrete

$M(t_1), M(t_2), \dots, M(t_n)$

Pro: Semplice. Contro: Ridondanza elevata, interpolazione temporale difficile.

2. **Space-Time Cube:** Ogni entità ha traiettoria 3D (x, y, t)

Entità E: $(x(t), y(t))$ per $t \in [t_0, t_1]$

Pro: Rappresentazione naturale di movimento. Contro: Non adatto a campi continui.

3. **Change-based:** Memorizzare stato iniziale + sequenza di cambiamenti

$M_0, \Delta_1, \Delta_2, \dots, \Delta_k$

$$M(t_k) = M_0 + \sum_{i=1}^k \Delta_i$$

Pro: Compatto. Contro: Query a epoca t richiede replay di tutti i cambiamenti.

4. **Modello funzionale:** Parametrizzare attributi come funzioni del tempo

$$z(x, y, t) = f(x, y, t; \theta)$$

dove θ sono parametri (ad esempio, coefficienti di polinomi temporali).

Esempio: Temperatura superficiale

Modello sinusoidale giornaliero + trend stagionale:

$$\begin{aligned} T(x, y, t) = & T_media(x,y) + A_giorno(x,y) \cdot \sin(2\pi t/24h + \phi_giorno) \\ & + A_anno(x,y) \cdot \sin(2\pi t/365d + \phi_anno) \\ & + T_trend \cdot t \end{aligned}$$

Parametri da stimare per ogni punto (x,y) :

- T_media : temperatura media
- A_giorno, ϕ_giorno : ampiezza e fase del ciclo giornaliero
- A_anno, ϕ_anno : ampiezza e fase del ciclo annuale
- T_trend : trend lineare (climate change)

Interpolazione spaziale dei parametri:

I parametri $\theta(x,y)$ variano spazialmente in modo continuo. Interpolazione:

- Kriging spazio-temporale
- Regressione geograficamente ponderata (GWR)
- Splines thin-plate

13.6.2 Animazione Cartografica e Visualizzazione Dinamica

Per comunicare efficacemente evoluzione temporale, animazioni cartografiche sono potenti.

Principi di design per animazioni:

1. **Identità costante:** Oggetti mantengono colore/simbolo attraverso il tempo
2. **Smooth transitions:** Interpolare gradualmente tra stati

3. **Controlli temporali:** Play/pause, speed control, timeline slider
4. **Reference frames:** Mostrare contesto statico (confini, topografia)

Interpolazione temporale:

Tra due snapshot t_1 e t_2 , interpolare linearmente:

$$M(t) = (1 - \alpha) \cdot M(t_1) + \alpha \cdot M(t_2)$$

$$\text{dove } \alpha = (t - t_1) / (t_2 - t_1)$$

Per campi scalari (es. temperatura), interpolazione diretta funziona.

Per entità discrete (es. veicoli), interpolazione della posizione:

$$P(t) = (1 - \alpha) \cdot P(t_1) + \alpha \cdot P(t_2)$$

Problemi dell'interpolazione lineare:

Per traiettorie curve o accelerate, interpolazione lineare produce artefatti (movimenti a scatti, inversioni di direzione).

Soluzioni avanzate:

1. **Interpolazione spline cubica:**

$$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

con vincoli di continuità in velocità (C^1) o accelerazione (C^2).

2. **Filtro di Kalman per smoothing:** Dato sequenza di posizioni rumorose, stimare traiettoria smooth che le spiega.
3. **B-splines razionali (NURBS):** Per traiettorie complesse con controllo locale.

13.6.3 Previsione e Mappe Probabilistiche

Per fenomeni futuri, le mappe devono rappresentare incertezza.

Mappe probabilistiche:

Invece di valori deterministici, rappresentare distribuzioni:

$$P(z \mid x, y, t)$$

Visualizzazione:

- Valor atteso (mean) + fasce di confidenza
- Heatmaps con opacità proporzionale a incertezza
- Ensemble di scenari

Esempio: Previsione diffusione inquinante

Modello di diffusione atmosferica:

$$\partial C / \partial t = D \nabla^2 C - v \cdot \nabla C + S - \lambda C$$

dove:

- $C(x,y,t)$: concentrazione
- D : coefficiente di diffusione
- v : campo vettoriale di vento
- S : sorgenti
- λ : tasso di decadimento

Parametri incerti (vento, D) → incertezza nella previsione.

Simulazione Monte Carlo:

1. Campionare N set di parametri dalla distribuzione di incertezza
2. Eseguire N simulazioni deterministiche
3. Per ogni punto e tempo, calcolare distribuzione di C

Visualizzazione: Percentili (10%, 50%, 90%) come curve isoline.

13.7 Big Data Geospaziali e Cloud Computing

13.7.1 Elaborazione Distribuita di Dati Cartografici

Dati satellitari moderni producono petabyte/anno. Elaborazione richiede infrastrutture distribuite.

Framework di calcolo distribuito:

1. MapReduce paradigm (Hadoop):

Map: $(key_1, value_1) \rightarrow list(key_2, value_2)$

Reduce: $(key_2, list(value_2)) \rightarrow list(value_3)$

Esempio: Calcolo NDVI (Normalized Difference Vegetation Index) su scena Sentinel-2:

Map:

Input: blocco di pixel (i,j)
 Leggi bande NIR e RED
 Calcola NDVI = (NIR - RED)/(NIR + RED)
 Output: (i,j, NDVI)

Reduce:

Input: tutti NDVI della tile
 Statistiche: media, std dev, istogramma
 Output: tile statistics

2. Spark (in-memory computing):

- Più veloce di Hadoop per workflow iterativi
- RDD (Resilient Distributed Datasets) mantengono dati in RAM
- Spark GeoSpatial: estensioni per dati spaziali

3. Dask (Python ecosystem):

- Parallelizzazione trasparente di NumPy, Pandas
- Lazy evaluation: costruisce grafo computazionale, esegue solo quando necessario
- Scalabile da laptop a cluster

Esempio Dask per raster processing:

```
import dask.array as da
import rasterio

# Aprire raster enorme (100 GB) come dask array
with rasterio.open('huge_raster.tif') as src:
    raster = da.from_array(src.read(1), chunks=(1024, 1024))

# Operazioni lazy (non eseguite immediatamente)
mean = raster.mean()
std = raster.std()
filtered = da.where(raster > mean + 2*std, 255, raster)

# Trigger computation
result = filtered.compute() # Esegue tutto in parallelo
```

13.7.2 Tiling e Indexing Spaziale

Per query efficienti su grandi dataset, indicizzazione spaziale è essenziale.

Quad-tree tiling (come discusso):

Ricorsivamente suddivide spazio in 4 quadranti. ID tile codificato come stringa:

Root: ""

Level 1: "0", "1", "2", "3" (NW, NE, SW, SE)

Level 2: "00", "01", "02", "03", "10", "11", ...

Z-order curve (Morton encoding):

Mappa 2D (x,y) → 1D z tramite interleaving bits.

Algoritmo:

$x = x_0x_1x_2\dots x_{n-1}$ (binary)

$y = y_0y_1y_2\dots y_{n-1}$

$z = y_0x_0y_1x_1y_2x_2\dots y_{n-1}x_{n-1}$

Esempio:

$x = 5 = 101_2$

$y = 3 = 011_2$

$z = 010111_2 = 23$

Proprietà: Punti vicini in 2D tendono ad avere z vicini → buona località per storage e query.

Hilbert curve:

Come Z-order ma con migliore locality preservation (curve spaziale continua).

R-tree:

Struttura dati ad albero per rettangoli (bounding boxes):

- Ogni nodo interno: bounding box che contiene tutti i figli
- Nodi foglia: oggetti effettivi
- Bilanciato: tutte le foglie a stessa profondità

Query range efficiente:

Per trovare tutti gli oggetti intersecanti regione Q:

function range_query(node, Q):

 if node.bbox does not intersect Q:

 return []

 if node is leaf:

 return [obj for obj in node.objects if obj intersects Q]

 else:

 results = []

 for child in node.children:

 results += range_query(child, Q)

return results

Complessità: $O(\log n + k)$ dove k è numero risultati.

13.7.3 Cloud-Optimized Geospatial Formats

Formati tradizionali (GeoTIFF singolo file enorme) non sono efficienti per cloud storage e streaming.

Cloud-Optimized GeoTIFF (COG):

GeoTIFF standard con:

1. **Tiling interno:** Invece di scanline, organizzato in tiles (es. 512×512)
2. **Overviews:** Piramidi pre-calcolate a risoluzioni inferiori
3. **HTTP Range Requests:** Possibile leggere solo tiles necessari senza scaricare tutto

Struttura file:

[Header: metadata, CRS, georeferencing]

[IFD 0: Full resolution tiles]

Tile 0,0

Tile 0,1

...

[IFD 1: Overview 1 (1:2)]

Tile 0,0

...

[IFD 2: Overview 2 (1:4)]

...

Beneficio: Per visualizzare regione piccola a zoom alto, leggere solo:

- Overview appropriato
- Tiles necessari

Invece di intero file (risparmio: 99%+ per regioni piccole).

Zarr format:

Array multi-dimensionali chunked su cloud storage.

data.zarr/

.zarray (metadata)

.zattrs (attributi)

0.0.0 (chunk)

0.0.1

0.1.0

...

Ogni chunk è file separato → parallelismo perfetto.

STAC (SpatioTemporal Asset Catalog):

Standard per cataloghi di dati spaziotemporali. JSON-based:

```
{
  "type": "Feature",
  "stac_version": "1.0.0",
  "id": "sentinel2_tile_33TUG_2023-03-15",
  "geometry": {...},
  "properties": {
    "datetime": "2023-03-15T10:20:31Z",
    "platform": "Sentinel-2A",
    "instruments": ["MSI"]
  },
  "assets": {
    "B04": {
      "href": "https://sentinel2.../B04.tif",
      "type": "image/tiff; application=geotiff"
    },
    "B08": {...}
  }
}
```

Permette discovery e accesso a dataset distribuiti.

13.8 Machine Learning per Cartografia Automatizzata

13.8.1 Deep Learning per Segmentazione Semantica

Classificazione pixel-by-pixel di immagini satellitari/aeree.

Architettura U-Net:

Encoder (downsampling):

Conv 3x3, ReLU

Conv 3x3, ReLU

MaxPool 2x2

[Ripetere con canali crescenti: 64→128→256→512]

Bottleneck:

Conv 3x3, ReLU

Conv 3x3, ReLU

Decoder (upsampling):

UpConv 2x2

Concatenazione con encoder layer (skip connection)

Conv 3x3, ReLU

Conv 3x3, ReLU

[Ripetere con canali decrescenti: 512→256→128→64]

Output:

Conv 1x1 (classificazione per pixel)

Softmax (probabilità per classe)

Skip connections: Permettono al decoder di usare features sia ad alto livello (semantiche) che basso livello (dettagli geometrici).

Training:

Dataset: Immagini con ground truth (pixel classificati manualmente) Loss function: Cross-entropy per pixel

$$L = -\sum_i \sum_c y_{ic} \log(\hat{y}_{ic})$$

dove:

- i: indice pixel
- c: classe
- y_{ic} : ground truth (one-hot)
- $\hat{y}_{ic}$: predizione (probabilità)

Ottimizzatore: Adam (adaptive learning rate)

Augmentation: Essenziale per generalizzazione

- Rotazioni random
- Flip orizzontale/verticale
- Variazioni di luminosità/contrasto
- Elastic deformations

Problemi e soluzioni:

1. **Class imbalance:** Alcune classi rare (es. edifici 5%, vegetazione 80%) Soluzione: Weighted loss, focal loss
2. **Boundary artifacts:** Bordi tra classi spesso classificati male Soluzione: Boundary-aware loss, post-processing con CRF

3. **Multi-scale objects:** Edifici piccoli e grandi Soluzione: Multi-scale inputs, atrous convolution

13.8.2 Object Detection per Feature Extraction

Rilevare oggetti specifici (edifici, veicoli, navi, etc.) con bounding boxes.

Architettura Faster R-CNN:

1. **Backbone CNN:** Estrae feature maps (es. ResNet-50)
2. **Region Proposal Network (RPN):** Propone regioni candidato
3. **ROI Pooling:** Estrae features per ogni regione
4. **Classification + Regression:** Classifica oggetto e raffina bounding box

Mask R-CNN: Estensione con segmentazione instance (maschera pixel per ogni oggetto).

YOLO (You Only Look Once): Approccio single-shot più veloce

- Divide immagine in griglia (es. 13×13)
- Ogni cella predice bounding boxes + classi
- Molto veloce (real-time) ma meno accurato su oggetti piccoli

Metriche di valutazione:

- **Precision:** $TP / (TP + FP)$
- **Recall:** $TP / (TP + FN)$
- **Average Precision (AP):** Area sotto curva precision-recall
- **Mean AP (mAP):** Media di AP su tutte le classi

Per essere considerata corretta, detection deve avere:

$IoU = \text{Area}(\text{prediction} \cap \text{ground_truth}) / \text{Area}(\text{prediction} \cup \text{ground_truth}) > \text{threshold}$

tipicamente threshold = 0.5 o 0.75.

13.8.3 Generative Models per Map Synthesis

Generare mappe realistiche automaticamente.

GAN (Generative Adversarial Network) per mappe:

- **Generator G:** Mappa rumore/sketch → mappa realistica
- **Discriminator D:** Distingue mappe reali da generate

Training avversariale:

$$\min_G \max_D \mathbb{E}[\log D(x)] + \mathbb{E}[\log(1 - D(G(z)))]$$

Applicazioni:

1. **Image-to-image translation:** Sketch $\rightarrow$ colored map, satellite $\rightarrow$ map, map $\rightarrow$ satellite

Pix2Pix architecture:

- Generator: U-Net
 - Discriminator: PatchGAN (classifica patch 70×70)
 - Loss: L1 (pixel-wise) + adversarial
2. **Super-resolution:** Low-res map $\rightarrow$ high-res map

ESRGAN (Enhanced Super-Resolution GAN):

- Rete generativa profonda con residual blocks
 - Upsampling progressivo (2x $\rightarrow$ 4x $\rightarrow$ 8x)
 - Perceptual loss (feature matching su VGG network)
3. **Style transfer:** Applicare stile cartografico a immagini

CycleGAN: Trasformazione unpaired (non servono coppie esatte)

- Due generatori: $G: X \rightarrow Y$, $F: Y \rightarrow X$
- Cycle consistency: $F(G(x)) \approx x$, $G(F(y)) \approx y$

13.9 Cartografia per Veicoli Autonomi**13.9.1 HD Maps (High Definition Maps)**

Mappe per guida autonoma richiedono precisione centimetrica e dettaglio estremo.

Contenuto di HD map:

1. **Road geometry:** Centerlines, boundaries, curvature (precisione < 10 cm)
2. **Lane markings:** Tipo (solida, tratteggiata), colore, larghezza
3. **Traffic signs:** Posizione 3D, tipo, orientazione
4. **Traffic lights:** Posizione, mapping a corsie controllate
5. **Semantic information:** Crosswalks, stop lines, speed limits
6. **3D features:** Poles, barriers, curbs per localizzazione

Livelli di dettaglio:

- **Geometric layer:** Polyline 3D con attributi
- **Semantic layer:** Classificazione elementi
- **Relational layer:** Connettività (quale corsia connette a quale)
- **Regulatory layer:** Regole (precedenza, velocità)

Formato OpenDRIVE:

Standard XML per road networks:

```
<road id="1" junction="-1" length="100.0">
  <planView>
    <geometry s="0.0" x="0.0" y="0.0" hdg="0.0" length="100.0">
      <line/>
    </geometry>
  </planView>
  <lanes>
    <laneSection s="0.0">
      <left>
        <lane id="1" type="driving" level="false">
          <width sOffset="0.0" a="3.5" b="0.0" c="0.0" d="0.0"/>
        </lane>
      </left>
      <center>
        <lane id="0" type="none" level="false"/>
      </center>
      <right>
        <lane id="-1" type="driving" level="false">
          <width sOffset="0.0" a="3.5" b="0.0" c="0.0" d="0.0"/>
        </lane>
      </right>
    </laneSection>
  </lanes>
</road>
```

13.9.2 Localizzazione Basata su Mappa

Il veicolo usa HD map + sensori per localizzarsi con precisione.

Particle Filter Localization:

Rappresenta belief sulla posizione come insieme di particelle:

$$S = \{(x_i, w_i) \mid i = 1, \dots, N\}$$

dove x_i è una posa (x, y, θ) e w_i è il peso (probabilità).

Algoritmo:

1. **Prediction:** Muovi particelle secondo modello di movimento

$$x'_i = x_i + \text{motion_model}(u, \text{noise})$$

2. **Update:** Peso particelle secondo match tra osservazioni e mappa

$$w_i = p(z | x_i, \text{map})$$

3. **Resampling:** Ricampiona particelle proporzionale a pesi

Observation model:

Veicolo scansiona ambiente (LiDAR) → point cloud Confronta con features nella HD map (poles, curbs)

$$p(z | x, \text{map}) = \prod_j p(z_j | x, \text{map})$$

Per ogni punto scansionato z_j , cercare corrispondenza nella mappa trasformata secondo posa x .

ICP (Iterative Closest Point):

Allinea point cloud con map:

1. Per ogni punto scansionato, trova punto più vicino nella mappa
2. Calcola trasformazione (R, t) che minimizza distanze
3. Applica trasformazione, ripeti

Converge a allineamento locale ottimale.

Accuratezza ottenibile:

Con HD map + LiDAR + IMU + GPS:

- Longitudinale: 5-10 cm
- Laterale: 5-10 cm
- Heading: 0.1-0.2°

Essenziale per lane-keeping preciso.

13.9.3 Dynamic Map Updates

HD maps devono essere aggiornate frequentemente (costruzione, incidenti, ecc.).

Crowdsourcing da flotta:

Migliaia di veicoli con sensori continuamente raccolgono dati.

Change detection:

Confrontare osservazioni recenti con mappa:

1. Allineare point cloud a mappa esistente
2. Identificare outliers (punti non spiegati da mappa)
3. Cluster outliers → potenziali nuovi oggetti o changes
4. Validare con multiple osservazioni indipendenti

Classificazione automatica di changes:

Machine learning per categorizzare:

- Construction (temporaneo)
- New permanent feature (aggiornare mappa)
- Parked vehicle (ignorare)
- Map error (correggere)

Pipeline di aggiornamento:

Veicoli → Raw data → Cloud processing → Change detection →
→ Human verification → Map update → Distribution → Veicoli

Latenza: ore-giorni (non real-time ma abbastanza rapido per essere utile).

Capitolo 14: Aspetti Giuridici e Standardizzazione

14.1 Standard Cartografici Internazionali

14.1.1 OGC (Open Geospatial Consortium)

Organizzazione che definisce standard aperti per dati geospaziali.

Standard principali:

1. **WMS (Web Map Service):** Serve immagini di mappe via HTTP

```
http://server/wms?SERVICE=WMS&REQUEST=GetMap
&LAYERS=roads,buildings&STYLES=&SRS=EPSG:4326
&BBOX=12.0,41.8,12.6,42.0&WIDTH=600&HEIGHT=400
&FORMAT=image/png
```

2. **WFS (Web Feature Service):** Serve dati vettoriali (GeoJSON, GML)

```
http://server/wfs?SERVICE=WFS&REQUEST=GetFeature
&TYPENAME=buildings&BBOX=12.0,41.8,12.6,42.0
```

3. **WCS (Web Coverage Service):** Serve dati raster (valori, non solo immagini)

4. **CSW (Catalog Service for the Web):** Discovery di dataset

GeoJSON format:

Standard per geometrie in JSON:

```
{
  "type": "Feature",
  "geometry": {
    "type": "Polygon",
    "coordinates": [[[12.0,41.8], [12.1,41.8], [12.1,41.9], [12.0,41.9], [12.0,41.8]]]
  },
  "properties": {
    "name": "Building A",
    "height": 25.5
  }
}
```

14.1.2 ISO Standards per Geomatics

ISO 19100 series: Framework completo per dati geospaziali

- **ISO 19111:** Spatial referencing by coordinates (CRS)
- **ISO 19115:** Metadata
- **ISO 19125:** Simple feature access (geometrie)
- **ISO 19136:** Geography Markup Language (GML)

Well-Known Text (WKT) for CRS:

Format testuale per definire CRS:

```
PROJCS["WGS 84 / UTM zone 33N",
  GEOGCS["WGS 84",
    DATUM["WGS_1984",
      SPHEROID["WGS 84",6378137,298.257223563]],
    PRIMEM["Greenwich",0],
    UNIT["degree",0.0174532925199433]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",15],
  PARAMETER["scale_factor",0.9996],
  PARAMETER["false_easting",500000],
  PARAMETER["false_northing",0],
  UNIT["metre",1]]
```

14.2 Proprietà Intellettuale e Licenze

14.2.1 Copyright su Dati Cartografici

Dati geografici possono avere copyright in molte giurisdizioni.

Protezione:

- **Database directive (EU):** Protegge investimento sostanziale in raccolta/organizzazione dati
- **Copyright:** Protegge espressione creativa (styling, generalizzazione)
- **Non protegge:** Fatti grezzi (coordinate di un punto)

Licenze open data:

1. **OpenStreetMap:** ODbL (Open Database License)
 - Share-alike: derivati devono usare stessa licenza
 - Attribution richiesta
2. **Creative Commons:**
 - CC-BY: Attribution only
 - CC-BY-SA: Attribution + Share-alike
 - CC-0: Public domain dedication
3. **Public domain:** Dati governativi spesso nel pubblico dominio (varia per paese)

14.2.2 Precisione e Responsabilità Legale

Errori cartografici possono avere conseguenze legali.

Casi di responsabilità:

- Confini proprietà errati → dispute legali
- Mappe navigazione errate → incidenti
- Mappe rischio (alluvione, etc.) errate → danni non prevenuti

Disclaimers tipici:

"This map is provided for general information purposes only. The provider makes no representations or warranties regarding accuracy. Users should verify information independently for critical applications."

Standard di accuratezza:

- **ASPRS (American Society for Photogrammetry and Remote Sensing):** Positional Accuracy Standards for Digital Geospatial Data
 - Classe I: $RMSE_x, RMSE_y \leq 1 \text{ m}$
 - Classe II: $RMSE \leq 2 \text{ m}$

○ ...

Capitolo 15: Futuro della Cartografia

15.1 Quantum Computing per Problemi Geospaziali

Algoritmi quantistici potrebbero rivoluzionare certe classi di problemi.

Ottimizzazione combinatoria: Trovare proiezione ottimale per regione irregolare

Quantum Annealing potrebbe esplorare spazio soluzione esponenzialmente più veloce.

Simulazione quantistica: Modelli fisici (atmosfera, oceani) con precisione molecolare.

15.2 Digital Twins Planetari

Repliche digitali complete della Terra aggiornate in tempo reale.

Progetto Destination Earth (EU):

Obiettivo: Simulazione completa del sistema Terra a risoluzione chilometrica, aggiornata continuamente con osservazioni.

Applicazioni:

- Previsioni meteo a lungo termine
- Impatto climate change
- Gestione risorse naturali
- Pianificazione urbanistica globale

15.3 Cartografia Extraterrestre

Con esplorazione spaziale intensificata, cartografia di Luna, Marte, asteroidi.

Sfide uniche:

- Assenza di datum consolidati
- Referencing basato su cratere/feature identificabile
- Rotazione non uniforme (es. Phobos, luna di Marte)
- Coordinate selenografiche vs areografiche

Standards emergenti:

IAU (International Astronomical Union) definisce sistemi di coordinate planetarie.

Conclusione

La cartografia matematica rappresenta una sintesi affascinante di geometria differenziale, analisi numerica, geodesia fisica e informatica. Dall'impossibilità fondamentale stabilita dal

Theorema Egregium di Gauss alle sofisticate pipeline di machine learning per generazione automatica di mappe, il campo continua ad evolversi.

Le tecnologie emergenti—GNSS multi-costellazione, LiDAR ad alta risoluzione, satelliti di osservazione terrestre con revisit giornaliero, quantum sensing—promettono accuratezze e risoluzioni impensabili anche solo un decennio fa.

Allo stesso tempo, le sfide fondamentali rimangono: la Terra è una superficie curva e qualsiasi sua rappresentazione planare comporta distorsioni. La scelta della proiezione appropriata resta un esercizio di compromesso guidato dall'applicazione specifica.

La democratizzazione dei dati geospaziali tramite iniziative come OpenStreetMap, combinata con potenti strumenti open-source (QGIS, GDAL, PROJ), ha reso la cartografia accessibile a un pubblico vastissimo. Ma con questa accessibilità viene la responsabilità: comprendere i fondamenti matematici e le limitazioni dei sistemi cartografici è più importante che mai.

In un'era di veicoli autonomi, realtà aumentata ubiqua, e modelli predittivi ad alta risoluzione, la cartografia non è più solo l'arte di disegnare mappe—è l'infrastruttura critica che connette il mondo fisico con i suoi modelli digitali.